



WEB BASED HOTEL MANAGEMENT SYSTEM FOR DREAM BEACH RESORT

P.T.S.S SWARNAMALI

BIT Registration No: R051992

Index No: 0519928

Supervisor: Mrs. Imeshika Sarathchandra

2009/ 2010



**This dissertation is submitted in partial fulfillment of the requirement of the
Degree of Bachelor of Information Technology (External) of the University of
Colombo School of Computing**

Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Signature of Candidate: *Sai Preethi*.....

Date: 20.10./.29/.09.2

Name of Candidate: P.T.S.S.Swarnamali

Countersigned by:

Signature of supervisor(s): *Dh*.....

Date: 10./.09./.2

Name(s) of Supervisor(s): *Tumeshika Sarathchandra*

Abstract

The developed system is a web based hotel Management System, which would help the administration process of the hotel. This system manages its resources effectively and improves the productivity and efficiency of the hotel using the world-wide information technology. This software provided the infomation about room availability, their features and rates to the customers around the world via attractive web interface integrated with company web site. The staff members of the hotel could also access the system depending on their authentication and authorization level. The provided system has the role of system administrator who will look into all the features of the system supporting fine execution of the system.

The existing manual system has many failures and limitations which affect overall hotel goals and objectives. The company management has found that is very hard to handle data manually when the data volume is high with acceleration of the company. it difficult to handle data accurately because of the data lost. Therefore this software is designed and developed considering basically the requirements of Dream Beach hotel.

The developed system has the ability to handle the online room reservation, cancel and modify the reservation, check-in and check-out functions of the guest, allocate room, issue guest bill, income analysis, customer and compannny employee profiles maintenance,company profile maintenance, generate various reports,define reference and master data, security and access control for the system. These facilities are provided in the system with user friendly interface, so authorized user can add, delete and update the entries of the system and handle all the transactions easily.

Object-Oriented methodologies were used in designing the system and Open source technologies were utilized in developing the system according to the client requirement. PHP as a programming language, MySQL was selected as the database software and apache was selected as web server for this system.

This document explains how the developed system would handle the above mentioned functional areas of a hotel, and how it would be helpful to improve the efficiency and productivity of the hotel.

Acknowledgements

I express my thanks to the Director and all the academic staff of University of Colombo School of computing, for designing this type of degree program that is very useful to us for build our career path

Mrs. K.D.I.U.Sarathchandra, the project supervisor would be offered my greatest gratitude for the guidance provided by her and support to get out of dead ends at times from the very beginning of the project. She has taken pain to go through the project and make necessary correction as and when needed.

I am especially indebted to Mr. Prasanna the client of the project who gave an enormous help to success my project. Specially for giving me time to requirement gathering, requirement analyzing and system testing.

I am grateful to entire lecture panel of Openarc School of business and technology for giving adequate knowledge to success my degree program.

I am also thankful to my family, friends and mates who have rendered their whole hearted support at all times for the successful completion of project.

Contents

Declaration	ii
Abstract.....	iii
Acknowledgements.....	iv
Contents	v
List of Figures	ix
List of Tables.....	xi
List of Acronyms	xii
Chapter 1 – Introduction	1
1.1. Motivation for Project.....	1
1.2. Scope of the Project.....	2
1.3. Objective of the Project	3
1.4. Dissertation Overview	3
Chapter 2 – Analysis.....	5
2.1. Similar Systems and Features.....	5
2.1.1. HOTSOFT-Hot Solution with a Soft Touch.....	5
2.1.2. OHMS - Online Hotel Management System.....	6
2.1.3. eZee Front Desk.....	7
2.2. Requirement Gathering.....	8
2.3. Functional Requirements	8
2.3.1. Hotel Data Maintenance.....	8
2.3.2. Online Room Reservation	8
2.3.3. Reservation Cost Analysis.....	8
2.3.4. Guest Check-in and Check-out Handling.....	9
2.3.5. Income Analysis	9
2.3.6. Employee Details Maintenance	9
2.4. Non-Functional Requirements	9
2.4.1. Security Requirements	9
2.4.2. Maintainability.....	9

2.4.3.	Meaningful Message	9
2.4.4.	Usability	9
2.5.	System Users	9
2.6.	Users Interactions with the System	10
	Chapter 3 – Design	12
3.1.	System Design.....	12
3.2.	Alternate Solution Evaluation	12
3.3.	Selected Solution Description/Justification	13
3.3.1.	Use Case Diagram	13
3.3.1.1.	Reference Data, System Execution and Reports Related Use Case Diagram.	14
3.3.1.2.	System Security/Access Control Related Use Case Diagram.....	15
3.3.2.	Use Case Narratives	16
3.3.3.	Class Diagram.....	24
3.3.4.	Activity Diagram	25
3.3.5.	Sequence Diagrams.....	26
3.3.5.1.	Make Reservation Related Sequence Diagram.....	26
3.3.5.2.	Modify Reservation Related Sequence Diagram	27
3.3.5.3.	Cancel Reservation Related Sequence Diagram.....	28
3.3.6.	Entity Relationship Diagram	28
3.4.	User Interface	29
3.4.1.	Evaluation Criteria of a Good User Interface	29
3.4.2.	System Navigation	30
3.4.3.	Interface Standard	30
3.4.4.	List of Input Screens	31
3.4.5.	List of Output Screens.....	32
3.4.6.	Sample User Interfaces	33
3.4.6.1.	Home Page of the Website	33
3.4.6.2.	Reservation Page of the Website	34
3.4.6.3.	Login Page.....	35
3.4.6.4	Main Interface of the System.....	36
3.4.6.5	Country Detail Maintenance.....	37
3.4.6.6.	Room Type Maintenance	38
3.4.6.7.	Company Profile	39
3.4.6.8.	Access Rights Interface.....	40

Chapter 4 – Implementation.....	41
4.1. Implementation Environment.....	41
4.1.1. Server side implementation	41
4.1.2. Client side implementation.....	42
4.2. Developments Tools	42
4.3. Hardware and software Requirements	42
4.3.1. Server side Hardware and Software Requirements.....	42
4.3.2. Client side Hardware and Software Requirements	43
4.4. Code and Module Structure	43
4.5. Sample Codes	45
4.5.1. Web Tier with PHP scripting.....	45
4.5.2. Class with attributes and functions	46
Chapter 5 – Evaluation.....	47
5.1. System Testing	47
5.1.1. Testing Phases	47
5.1.2. Test Plan.....	47
5.1.3. Test Cases.....	48
5.2 User Evaluation.....	54
Chapter 6 – Conclusion.....	57
6.1. Critical Assessment of Project	57
6.2. Future enhancement of the project	57
6.3. Lessons Learnt.....	58
6.4. Conclusion.....	58
References	59
Appendices	60
Appendix A- System Documentation.....	60
Appendix B- Design Documentation	62
Appendix C- User Documentation.....	66
Appendix D- Managements Reports	75
Appendix E- Test Results	79

Appendix F- Code Listing	83
Appendix G- Client Certificate	116
Glossary and Index	117

List of Figures

Figure 3.1	Use Case Diagram (Reference Data, System Execution and Reports).....	14
Figure 3.2	Use Case Diagrams (System Security/Access Control).....	15
Figure 3.3	Class Diagram.....	24
Figure 3.4	Activity Diagram.....	25
Figure 3.5	Sequence Diagram (Make Reservation).....	26
Figure 3.6	Sequence Diagram (Modify Reservation).....	27
Figure 3.7	Sequence Diagram (Cancel Reservation).....	28
Figure 3.8	Entity Relationship Diagram.....	29
Figure 3.9	List of input screens.....	31
Figure 3.10	List of output screens.....	32
Figure 3.11	Home Page of the System.....	33
Figure 3.12	Reservation Page of the Website.....	34
Figure 3.13	Login Page.....	35
Figure 3.14	Unauthorized login Attempt.....	35
Figure 3.15	Main Interface of the System.....	36
Figure 3.16	Country Details Maintenance.....	37
Figure 3.17	Room Types Maintenance.....	38
Figure 3.18	Company Profile.....	39
Figure 3.19	Access Rights Interface.....	40
Figure 4.1	Structure of the files and folders Dreambeach.....	43
Figure 5.1	User Evaluation Form.....	54
Figure 5.2	User Evaluation Report.....	56
Figure A.1	Apache Logo.....	60
Figure A.2	PHP Info Header.....	61
Figure C.1	Dream Beach Home Page.....	66
Figure C.2	System Login Page.....	67

Figure C.3	Main Modules of the System.....	67
Figure C.4	Define User Group (Access /Security Module).....	68
Figure C.5	Group Access Rights (Access /Security Module).....	69
Figure C.6	People Profile (Personal Management Module)	70
Figure C.7	Group User (Access/Security Module)	71
Figure C.8	Change Password (Access/Security Module)	71
Figure C.9	Profile Information (Company Profile)	72
Figure C.10	Define Title (Reference Data Module)	73
Figure C.11	Room Information (Master Data Module)	74
Figure D.1	Revenue Reports (Information Center).....	75
Figure D.2	Revenue Report as a Bar Chart.....	76
Figure D.3	Revenue Report as a Line Chart.....	76
Figure D.4	Guest Arrival List.....	77
Figure D.5	Guest Departure List.....	77
Figure D.6	Reservation Cancellation.....	78

List of Tables

Table 2.1	System Reference Data, System Execution and Reports.....	10
Table 2.2	Access Control Maintenance.....	11
Table 4.1	Server Side Hardware and Software Requirements.....	42
Table 4.2	Client Side Hardware and Software Requirements.....	43
Table 5.1	Test Case Descriptions –Login Page.....	48
Table 5.2	Test Case Descriptions – Room Type.....	48
Table 5.3	Test Case Descriptions – Room Pricing.....	49
Table 5.4	Test Case Descriptions – People Profile.....	50
Table 5.5	Test Case Descriptions – User Group.....	52
Table E.1	Test Case Descriptions – Define Country.....	79
Table E.2	Test Case Descriptions – Accommodation Type.....	80
Table E.3	Test Case Descriptions – Floor Type.....	81
Table E.4	Test Case Descriptions – Company Profile.....	82

List of Acronyms

Term	Description
GPL	General Public License
OS	Operating System
RUP	Rational Unified Process
PHP	PHP Hypertext Preprocessor
UI	User Interface
UML	Unified Modeling Language
WWW	World Wide Web
HTML	Hyper Text Transfer Protocol
IE	Internet Explorer
HMS	Hotel Management System
SMTP	Simple Mail Transfer Protocol
SVGA	Super Visual Graphics Array
GUI	Graphical User Interface
IMAP	Internet Message Access Protocol
CSS	Cascade Style Sheet
JS	JavaScript
MS	Microsoft
RAM	Random Access Memory
DB	Data Base
XML	Extensible Markup Language
HTTP	Hyper Text Transfer Protocol

Chapter 1 – Introduction

The proposed web based solution is developed for Dream Beach hotel to resolve current difficulties of the hotel. This system will facilitate the hotel to perform system transactions efficiently by eliminating all time consuming procedures and provide better services for their customers as well as the system users. This chapter will describe an overall idea about hotel management system for Dream Beach hotel.

1.1. Motivation for Project

Currently they are managing their system using MS EXCEL worksheet. Several problems are being faced by the organization, when providing services using the MS EXCEL Package. The company management has found that it is very hard and problematic to handle data manually when the data volume is high with the acceleration of the company

It's difficult to handle data accurately because of the data lost, viewing by unauthorized users of the system.

System users feel that it is difficult to manage data using Ms Excel Package. (Manage reservation, monitor revenue, room planning, meal planning and pricing plans etc...)

Getting information of the guest is difficult. (To know which guest is in which room, check-in, check-out details, payment & guest history etc...)

Getting information of the hotel is difficult. (To know the available of the hotel, availability of rooms and room rates etc...)

It takes more time for the reservation process. Thus performance and efficiency of the current system is slow.

Generating management report is time consuming and some are impractical.

Therefore they have identified that it is more useful to have an attractive website to compete with others in the market. So that they can provide up-to-date information about the hotel in no time, to the entire world and attract the interest of foreign guests. Thus the motivation for doing this project is to convert the manual system as a web based system including the company website. The organization would gain more

advantages by using a web based system along with the website, rather than using a standalone system.

1.2. Scope of the Project

The scope of the project has been covered by providing facility for system users and customer by eliminating difficulties mentioned in 1.1.

Facilities for system users

- Reference data maintenance
- Master data maintenance
- Online room reservation
- Allocate rooms
- Reservation cost analysis
- Guest check-in and check-out handling
- Income analysis
- Company and employee profile maintenance
- Security / access control for the system
- Generate system reports

Facilities for guests

- Display information about hotel and rooms.
- Allow the guests to view the price lists of rooms, food, and beverages through internet.
- Facility for guest to view available rooms according to the selected date.
- Facility for guests to reserve and cancel his/her own bookings.
- Maintain guest profile.

System provides the facility to define all the reference data and master data needed for the execution of above functionalities. The facility of maintaining access privileges for the different levels of users is provided in the system.

1.3. Objective of the Project

- The main objective of proposed project is to help the organization in automating the entire manual processing of the existing system.
- Provide a useful, highly available, platform independent web based hotel management system for the client, with website containing details about hotel.
- Provide way of maintaining up-to-date information of hotel so that it will increase the accuracy.
- To provide the best solution to this hotel with both internal and external operations such as front office and back office operations.
- Provide security feature like creating system users and assigning privileges to users of the system.
- Provide best facilities for the guest to view the up-to- date price list of the hotel rooms, their availability, and other details through the website.
- Improve efficiency of manual system.

1.4. Dissertation Overview

Chapter 2 – Analysis

Describes about the requirement gathering and analyzing techniques that had been used in the project

Chapter 3 – Design

Describes about the system designing with reference to certain design diagrams and user interactions with system with reference to several selected user interfaces and their functionalities.

Chapter 4 – Implementation

Discuss about the implementation environment and techniques. Development tools used the project and the reasons why they have been used for the implementation are also described. Code and module structures are explained with provided samples.

Chapter 5 – Evaluation

Describes about the various testing strategies performed for the validation and verification of the system

Chapter 6 – Conclusion

Concludes the dissertation by summarizing the achievements gained during the project while discussing about the expectable future enhancements can be done for the system

References

Acknowledgement of the work of others and WWW resources, which have been referred in order to get background information are included in this section

Appendices

Appendix A – Design Documentation

Appendix B – System Documentation

Appendix C – User Documentation

Appendix D – Management Reports

Appendix E – Test Results

Appendix F – Code Listing

Appendix G – Client Certificate

Chapter 2 – Analysis

This chapter contains inception phase of the RUP. During the inception phase, establish the business case for the system and identify the project scope. To accomplish this must identify all external entities with which the system will interact (actors) and define the nature of the interaction. Objectives of this phase:

- Establish project's scope and boundary conditions.
- Estimating the overall cost and schedule for the entire project.
- Estimating potential risks.

The outcome of the inception phase is:

- A general vision of the core project requirements
- Project Plan
- Initial risk assessment

2.1. Similar Systems and Features

2.1.1. HOTSOFT-Hot Solution with a Soft Touch [WWW4]

HOTSOFT is designed to suit every facet and need of the hotel industry enabling to provide effective, efficient and chivalrous service to guests, always.

HOTSOFT provide easy and practical solutions to common problems in day to day running of hotels. This hotel software that provides total control over the entire operations of the hotel is programmed to make life comfortable for the guests and easier for the hoteliers.

Modules of this system

- Front Office Management

The Front Office Management System is a multi featured module designed specifically for limited service independent properties and hotel chains. Its large, easy to read display gives quick to front desk functions with a touch screen graphic user interface. Simple user interface and online help make

training new staff quicker and easier. Features include reservations, front desk, cashier, housekeeping, night audit, system setup and over 100 standard reports.

- Restaurant Management System

Intuitive and proactive screens give the product extreme flexibility in various areas of operations viz. fast food outlets, remote captain order etc. This system provides an integrated F&B module and materials management module. The F&B module allows for both Issue Based Costing and Recipe Based Costing.

- Bar & Beverages Management System

The applications here are configurable to user specific requirements making it easy to use and to obtain fast, accurate, up to the minute information for simple and complex environments.

- Asset Management

The product has an impeccable record of proven efficiency, having been tried, tested and implemented successfully in a wide range of hospitality enterprises.

- CRS (Central Reservation System)

Through the CRS system, a single reservation is required for guest room/ function space sales for a chain of hotels on same/different days. Based on the availability, the user can directly reserve his/her room/function space. User will get confirmation through SMS alert and Email messages. Reservation advance is tightly linked through the payment gateway system. Bank transfer facility (e-Banking) also provided on this system. Online data transfer to the local premises of the Hotel through a Web service is an additional feature of the system.

2.1.2. OHMS - Online Hotel Management System [WWW9]

The system will help the Hotel Riviera to operate efficiently by eliminating all the time consuming procedures and provide a better service to their customers as well as for the employees in the company.

Two Main Parts of this system

- Web Based Reservation System (Customer's Perspective)
- Hotel Reservation System (Administration Perspective)

Features of the New System

- Handling Customer Details
- Reservations
- Cancellations
- Handling Rooms Details
- Making Payments
- Features like Update, Searching, Deleting, Editing Record etc...
- Generating different kinds of report.
- Customers can view videos of hotel rooms with audio explanations.
- Customers make reservations, enquiries and cancellations via online.

2.1.3. eZee Front Desk [WWW6]

eZee Front Desk is the modern solution which has whole range of integrated modules to cover every aspect of property management. The generalized version of eZee Front Desk Hotel reservation software is wide accepted worldwide with due to its state-of-art technology and extremely easy to use in nature.

Features of this system

- Check In, Check Out, Reservation
- Group Management
- Travel Agent and Corporate Accounts
- Night Audit
- House Keeping and Maintenance
- Mini-Bar
- Guest Relationship Management
- Cashiering
- Web Based Reporting
- Banquet Management

- Payroll and HR

2.2. Requirement Gathering

Requirement gathering and analyzing is one of the key feature to make the whole software development process a success. It's represent the interests and expectations of customers and users. In order to analyze requirements one need to gather them correctly and the whole project will become meaningless if the client's requirements are not gathered, analyze and defined accurately.

There are many ways to gather requirements such as interviews, questionnaires, observation, etc.

Method, which has been used to gather requirements for the proposed system, is conducting interviews. Most of the details were gathered by interviewing the system users. Details of the company reports and company documents were gathered for further inspections.

2.3. Functional Requirements

2.3.1. Hotel Data Maintenance

The system would provide the facility for the user of the hotel to maintain its room, room type wise(single,double,family).Pricing can be varied according to distinguished features of rooms(ac,non-ac) and define all the reference data needed for the execution of functionalities of the system.

2.3.2. Online Room Reservation

The system would provide the facility for the guest to reserve their room online. The system would facilitate to view room rates, food, and beverages, and available room according to the selected date when reservations are done via internet

2.3.3. Reservation Cost Analysis

The system would be able to calculate total cost of the reservation according to the room charges, food and beverage charges, no of days etc.

2.3.4. Guest Check-in and Check-out Handling

The system would be able to check-in guest with prior reservation and check-out guest.

2.3.5. Income Analysis

The system would be able to view a summary of income earned within a particular period of time.

2.3.6. Employee Details Maintenance

The system would provide the facility of maintaining employee information along with their employment status in the company.

2.4. Non-Functional Requirements

2.4.1. Security Requirement

The system should be highly secured so encrypted passwords and session management is used to avoid unauthorized access and misuse of the proposed system.

2.4.2. Maintainability

System should be easy to maintain because, business of the new proposed system is updated day to day.

2.4.3. Meaningful Message

System should present meaningful messages to user when prompting alerts or errors.

2.4.4. Usability

The proposed system should be easy to learn and include user manual which can guide and understand the system easily.

2.5. System Users

Following are the system user who would be considered as the actors in the use case diagram.

- System Administrator

- Normal User
- Front Officer
- HR Manager
- Room Manager
- Account Manager
- Company Management
- Food & Beverage Manager
- Guest

2.6. Users Interactions with the System

The system users and their functionalities with the system are described in the **Tables 2.1 and 2.2**. They have been explained using two use case diagrams (**Figure 3.1 and Figure 3.2**).

	Actor(s)	Function
System Reference Data, System Execution & Reports		
U1.1	Front Officer/Guest	Check Room Availability
U1.2	Front Officer/Guest	Make Reservation
U1.3	Front Officer/Guest	Modify Reservation
U1.4	Front Officer/Guest	Cancel Reservation
U1.5	Guest	View Rates/ Facilities
U1.6	Guest	View Meal Plans
U1.7	Guest	Login
U1.8	Guest	Get Hotel Information
U1.9	Guest	Check-in
U1.10	Guest	Check-out
U1.11	Guest	Make Payment
U1.12	Front Officer	Room Allocation
U1.13	Front Officer	Room Deallocation
U1.14	Front Officer	View Reservation
U1.15	Front Officer	Issue Bill
U1.16	Account Manager	Income Analysis
U1.17	Account Manager	Generate Reports
U1.18	Company Management	Update Company Profile
U1.19	Company Management	Maintain Room Type
U1.20	Company Management	Maintain Foods and Beverage
U1.21	Company Management	Maintain Room Details
U1.22	Company Management	Maintain Meal Plans
U1.23	Company Management	Update Room Rates/ Facilities
U1.24	Company Management	Maintain Accommodation Type
U1.25	Company Management	Maintain Floors
U1.26	HR Manager	Register Employee
U1.27	HR Manager	Update Employee Status
U1.28	HR Manager/Normal User	Update Employee Profile
Table 2.1 System Reference Data, System Execution and Reports		

	Actor(s)	Function
Access Control Maintenance		
U2.1	System Administrator	Update User Group
U2.2	System Administrator	Grant Revoke User Access
U2.3	System Administrator	Create User
U2.4	Normal User	Login
U2.5	Normal User	Change Password
Table 2.2 Access Control Maintenance		

Chapter 3 – Design

3.1. System Design

This chapter describe elaboration phase of the RUP. The purpose of the elaboration phase is to accomplish the functionalities discussed in inception phase, Capture remaining requirements and eliminate the highest risk elements of the project.

Object Oriented System Design techniques were chosen as the main system design methodology & the UML diagrams were used for illustrating them. In UML diagrams, the Use Case diagrams, Class diagrams, Sequence diagrams, Activity diagrams and collaboration diagrams were used to create the design of the proposed system.

3.2. Alternate Solution Evaluation

Why I did not use windows application?

This solution was required by hotel management to serve their customers worldwide and maximize their resource utilization providing up to date visibility across the hotel assets. In order to fulfill above requirements we need to make this solution accessible by both customers and staff. Windows application will provide this functionality but they need to be previously installed in their respective location in order to access the system which is overhead for the both local and foreign customers.

Why I did not choose stand alone web solution?

Current system was mostly manual so that communication between the customers and hotel staff is minimum till they reach their hotel therefore hotel needs to make their solution to be available to the clients .Stand alone application will not be accessible to the customers around the world which will not be advantageous to both customers and staff members of the hotel so that stand alone web solution was dropped.

Hotel also needs to make the customers details available consecutively minimize customer dissatisfaction by informing ahead about any environmental or asset inconvenience. This solution should also promote their business by playing the role of a customer awareness program.

Why I did not choose ASP.NET, SQL server?

Hotel management required this solution to be cost effective ASP.NET and SQL server technologies consumes huge licensing cost therefore I developed the system using PHP and MySQL. Platform independence was also a major requirement specified by the client, even though ASP.NET and SQL server supports any platform most of the time it is well integrated with Microsoft technologies.

3.3. Selected Solution Description/Justification

This major goal of this product is to expand the business and meet increasing demand from the customers spread around the world. The hotel management needs to make their resource information available so that customers make their own choice on traveling periods, meal requests and facilities. Therefore the developed system should maximize customer satisfaction by updating them prior to their arrival at the hotel. In response to above requirement I selected web solution.

Hotel management required the solution to be cost effective therefore I choose PHP as front end technology and MySQL to develop back end database. Both PHP and MySQL are open source technologies and integrate well with each other. PHP and MySQL are platform independent technologies which would perfectly execute with mostly used operating systems and popular browsers.

Another overhead of the current system was that it includes huge time wastage on allocation process since all the allocation details were captured using excel sheets. This issue was recognized and addressed by introducing the search functionality and providing MySQL database which minimize data redundancy and faster response time.

Ajax technology was used to provide the customers with a browser which is both attractive and highly responsive.

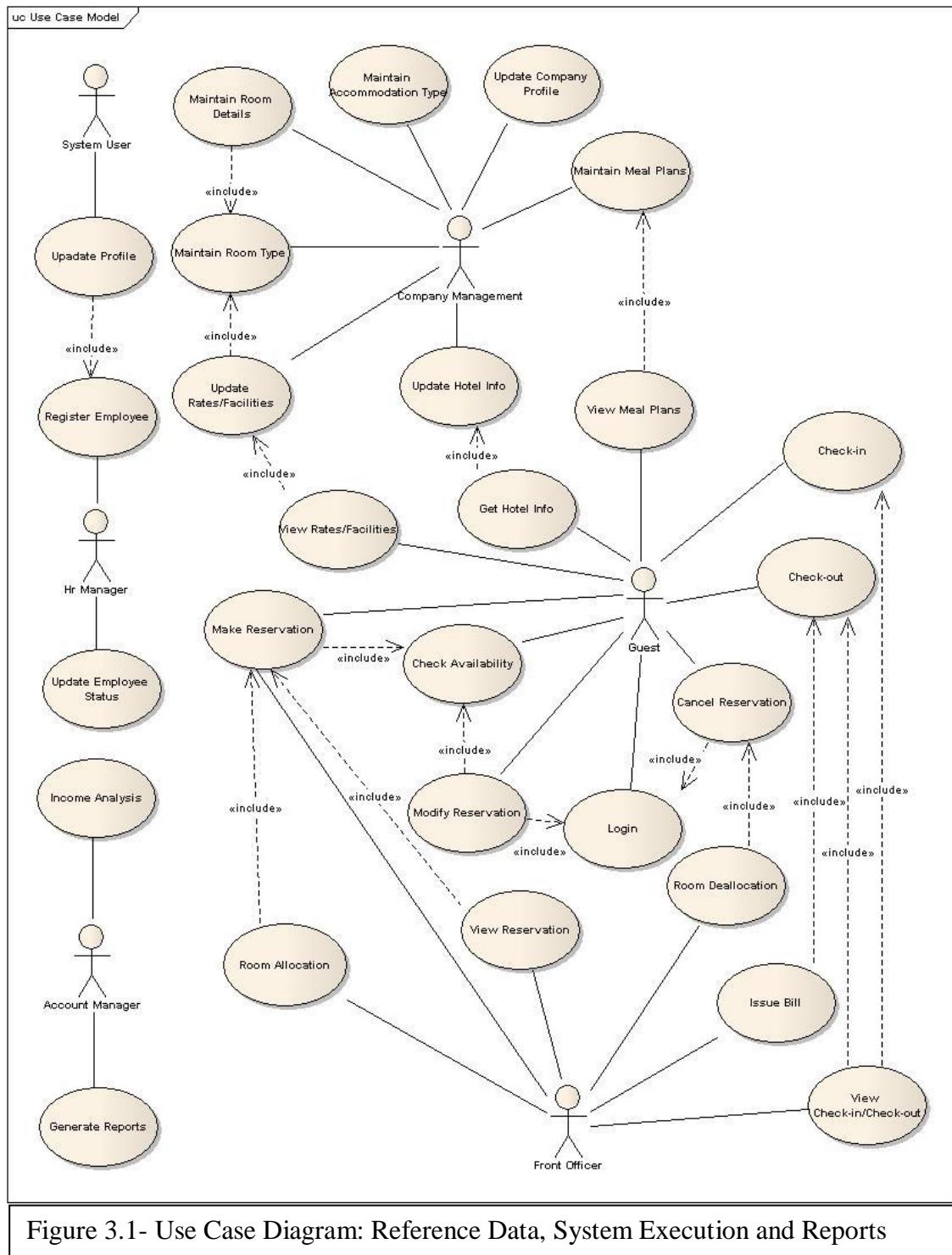
3.3.1. Use Case Diagram

The following diagrams display the functionality and the communication between the direct users of the system.

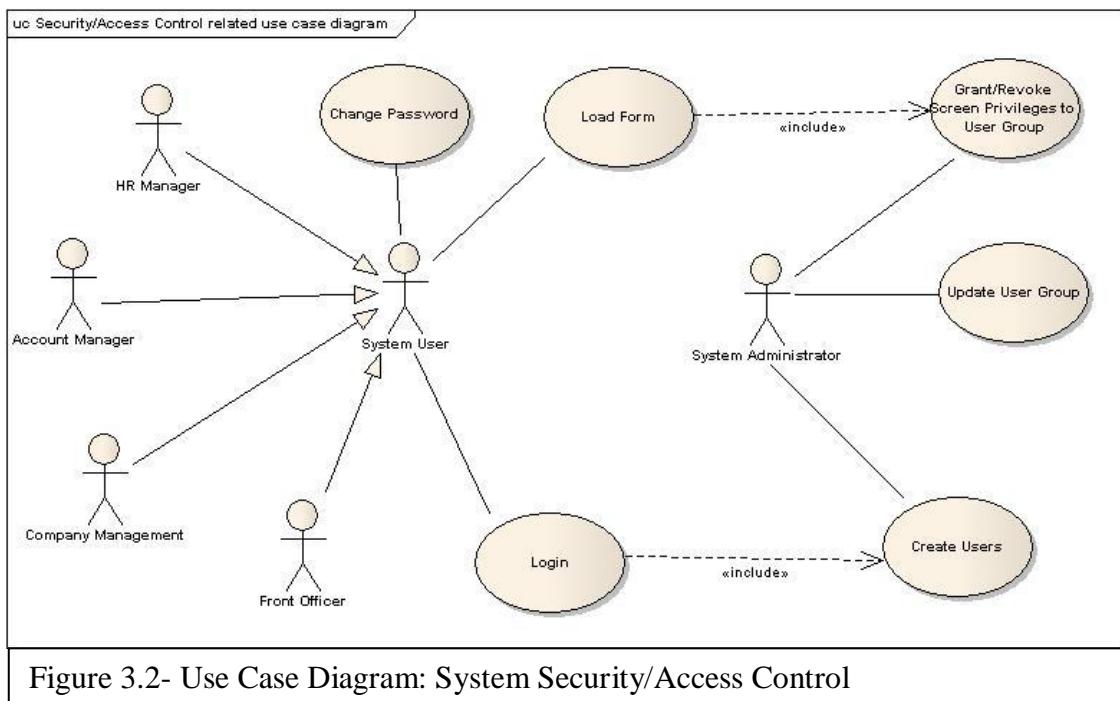
The first diagram (**Figure 3.1**) describes how all the reference data is handled during the system execution. It will provide the overview of all the functionalities of the system such as report generation, resource management and Customer communication.

The second diagram describes (**Figure 3.2**) process of authorization and authentication in the system.

3.3.1.1 Reference Data, System Execution and Reports Related Use Case Diagram



3.3.1.2 System Security/Access Control Related Use Case Diagram



3.3.2. Use Case Narrative

ID	U1.1
Name	Check Room Availability
Brief Description	Check availability of rooms according to the selected date when reservations are done via internet
Pre-conditions	Valid Arrival date and departure date were valid user inputs
Basic flow of events	<ol style="list-style-type: none"> 1. Check the database against the duration for each room type
Alternative flows	<ol style="list-style-type: none"> 1. Date in the past has been entered display the message containing the error 2. Arrival date is after the date of departure the display the message containing the error
Post- conditions	Display the result containing available no of rooms against room types and duration
Actors	Front officer, Guest
Related Use Cases	Make Reservation, Modify Reservation

ID	U1.2
Name	Make Reservation
Brief Description	Reserves a room for a hotel guest
Pre-conditions	Valid personal information and credit card details were user inputs
Basic flow of events	<ol style="list-style-type: none"> 1. Check the user has been entered valid personal information such as NIC/Passport, email and user name 2. Check the user has been entered valid credit card number
Alternative flows	<ol style="list-style-type: none"> 1. If NIC/Passport, email and user name are invalid display the message containing the error 2. If the credit card is not valid display the message containing the error
Post- conditions	The reservation created for the guest
Actors	Front officer, Guest
Related Use Cases	Check Room Availability, Make Reservation, Modify Reservation, Cancel Reservation

ID	U1.3
Name	Modify Reservation
Brief Description	Modify reservation reserved by guest
Pre-conditions	<ol style="list-style-type: none"> 1. Guest has logged in to guest control panel 2. He has added successful reservation
Basic flow of events	Guest access the reservation and modify
Alternative flows	<ol style="list-style-type: none"> 1. If the credit card is not valid display the message containing the error
Post- conditions	System has properly modified reservation
Actors	Front officer, Guest
Related Use Cases	Check Room Availability, Make Reservation, Modify Reservation, Cancel Reservation, Login

ID	U1.4
Name	Cancel Reservation
Brief Description	Cancel reservation reserved by guest
Pre-conditions	<ol style="list-style-type: none"> 1. Guest has logged in to guest control panel 2. He has added successful reservation
Basic flow of events	Guest access the reservation and cancel
Alternative flows	If the date difference between system date and arrival date less than 5 days , guest not allowed cancel the reservation
Post- conditions	System has properly modified reservation
Actors	Front officer, Guest
Related Use Cases	Make Reservation, Cancel Reservation, Login

ID	U1.5
Name	View Rates/ Facilities
Brief Description	Display the rates and facilities available in the hotel.
Pre-conditions	Guest has successfully accessed the system.
Basic flow of events	Display Updated Rates/Facilities
Alternative flows	<ol style="list-style-type: none"> 1. Check the system for all the details regarding the room types, rates and facilities. 2. Display the produced result in the web browser.
Post- conditions	Guest has a room details available in his browser.
Actors	Guest
Related Use Cases	Update Rates/ Facilities

ID	U1.6
Name	View Meal Plans
Brief Description	Display the meal rates and type of meals available in the hotel.
Pre-conditions	Guest has successfully accessed the system and requested the details of the meals.
Basic flow of events	<ol style="list-style-type: none"> 1. Check the system for the meal type's available, rates and special offers. 2. Display the result.
Alternative flows	
Post- conditions	Guest has a meal details available in his browser.
Actors	Guest
Related Use Cases	Update Meal Plans

ID	U1.7
Name	Login
Brief Description	Authenticate users to the system by validating their input and provide the browser interface depending on their level of authorization.
Pre-conditions	Users should have valid user name and password.
Basic flow of events	User enters his or her username and password.
Alternative flows	User enters an invalid credentials and system displays the error message.
Post- conditions	User can access successfully their control panel.
Actors	Front officer, Guest
Related Use Cases	Create User

ID	U1.8
Name	Get Hotel Information
Brief Description	Get the details of the hotel. (Location, year of establishment and etc.)
Pre-conditions	User has successfully accessed the system.
Basic flow of events	User requests the detail of the hotel.
Alternative flows	
Post- conditions	User can view successfully loaded hotel information in his browser.
Additional Notes	
Actors	Front officer, Guest
Related Use Cases	Update Hotel Information

ID	U1.9
Name	Check-in
Brief Description	Guest has arrived into hotel and informed the staff about his arrival and allocation details at reception.
Pre-conditions	Front officer has successfully logged into the system.
Basic flow of events	User selects the specific allocation details. Check the guests passport details NIC with the allocation If valid change the status of the allocation to check in.
Alternative flows	If the allocation details and guest details are found not matching if the resources are available make a new allocation.
Post- conditions	System has the arrived guests allocation status changed to check in.
Actors	Front officer
Related Use Cases	

ID	U1.10
Name	Check-out
Brief Description	Deallocates the resources held by the particular customer after bill is paid and customer is leaving the hotel.
Pre-conditions	Respective front officer has a successfully loaded system in his computer.
Basic flow of events	Front officer selects the specific allocation using the bill and change the status of the allocation to checkout.
Alternative flows	
Post- conditions	Allocation is in the status of check out.
Actors	Guest
Related Use Cases	Check-in

ID	U1.11
Name	Make Payment
Brief Description	Front officer selects the allocation and generate the total bill.
Pre-conditions	Customer has requested the bill.
Basic flow of events	<ol style="list-style-type: none"> 1. User selects the allocation and requests the bill from the system. 2. System generates the printable bill including the payment details and charges.
Alternative flows	

Post- conditions	Front officer has printout of the bill.
Actors	Guest
Related Use Cases	Check-out

ID	U1.12
Name	Room Allocation
Brief Description	Front officer generates the report on occupied rooms and change the status of the unoccupied rooms to free.
Pre-conditions	
Basic flow of events	<ol style="list-style-type: none"> 1. Room is not occupied. 2. Set the status of the room to free.
Alternative flows	
Post- conditions	All unoccupied rooms are available for the next allocation.
Actors	Front officer
Related Use Cases	Make Reservation

ID	U1.13
Name	Room Deallocation
Brief Description	Customer has made his total payment. Room is in the perfect condition.
Pre-conditions	Front officer has successfully logged into the system
Basic flow of events	<ol style="list-style-type: none"> 1. Front officer go for the room management page. 2. Mark the room as available for next allocation.
Alternative flows	
Post- conditions	Room is marked available and added to the number of available room's category.
Actors	Front officer
Related Use Cases	Room Allocation

ID	U1.14
Name	View Reservation
Brief Description	Front officer views the reservations for the hotel room types
Pre-conditions	Front officer has successfully logged into the system.
Basic flow of events	Front officer access the view reservation location.
Alternative flows	
Post- conditions	Front officer has all the reservation details available in his browser.
Actors	Front officer
Related Use Cases	Make Reservation

ID	U1.15
Name	Issue Bill
Brief Description	Customer has requested his bill.
Pre-conditions	Front officer has successfully logged into the system.
Basic flow of events	User enters the customer id.
Alternative flows	

Post- conditions	System will display and print the bill when requested.
Actors	Front officer
Related Use Cases	Check –out

ID	U1.16
Name	Income Analysis
Brief Description	Account Manager analyze income of the hotel
Pre-conditions	Account Manager has successfully logged into the system.
Basic flow of events	Select date or date range
Alternative flows	
Post- conditions	Account Manager has all the income details available in his browser.
Actors	Account Manager
Related Use Cases	Issue Bill

ID	U1.17
Name	Generate Reports
Brief Description	Report generation functionality to support the management decision making of the hotel.
Pre-conditions	Account manager has successfully logged into the system
Basic flow of events	Manager makes a request from the system about specific details.
Alternative flows	
Post- conditions	Manager has a successfully loaded report on his browser window.
Additional Notes	
Actors	Account Manager
Related Use Cases	

ID	U1.18
Name	Update Company Profile
Brief Description	Company Management updates the company profile
Pre-conditions	Company Management has successfully logged into the system.
Basic flow of events	Modify the existing profile
Alternative flows	
Post- conditions	Display modifying status
Actors	Company Management
Related Use Cases	

ID	U1.19
Name	Maintain Room Type
Brief Description	Company Management maintains room types of hotel
Pre-conditions	Company Management has successfully logged into the system.
Basic flow of events	<ol style="list-style-type: none"> 1. Add new room type 2. View existing room type 3. Modify/Delete existing room type
Alternative flows	<ol style="list-style-type: none"> 1. If the room type code already exist in the database do not allow to add

	2. If the room type code is used in transactions do not allow deleting
Post- conditions	Display saved details
Actors	Company Management
Related Use Cases	

ID	U1.20
Name	Maintain Foods and Beverage
Brief Description	Company Management maintains foods and beverage of hotel
Pre-conditions	Company Management has successfully logged into the system.
Basic flow of events	<ul style="list-style-type: none"> 1. Add new foods/beverage code 2. View existing foods and beverage 3. Modify/Delete existing foods and beverage
Alternative flows	<ul style="list-style-type: none"> 1. If the foods/beverage code already exist in the database do not allow to add 2. If the foods/beverage code is used in transactions do not allow deleting
Post- conditions	Display saved details
Actors	Company Management
Related Use Cases	

ID	U1.21
Name	Maintain Room Details
Brief Description	Company Management defines rooms of the hotel
Pre-conditions	Company Management has successfully logged into the system.
Basic flow of events	<ul style="list-style-type: none"> 1. Add new Room Details 2. Modify Room Details 3. Delete Room Details
Alternative flows	<ul style="list-style-type: none"> 1. If the Room No already exist, do not allow to add 2. If the No is used in transactions do not allow deleting
Post- conditions	Display saved details
Actors	Company Management
Related Use Cases	

ID	U1.22
Name	Maintain Meal Plans
Brief Description	Company Management defines Meal Plans of the hotel
Pre-conditions	Company Management has successfully logged into the system.
Basic flow of events	<ul style="list-style-type: none"> 1. Add new Meal Plans 2. Modify Meal Plans 3. Delete Meal Plans
Alternative flows	<ul style="list-style-type: none"> 1. If the Room Meal Plans code already exist, do not allow to add 2. If the Meal Plan code is used in transactions do not allow deleting
Post- conditions	Display saved details
Actors	Company Management
Related Use Cases	

ID	U1.23
Name	Update Room Rates/ Facilities
Brief Description	Company Management defines Room Rates and Facilities for hotel
Pre-conditions	Company Management has successfully logged into the system.
Basic flow of events	<ol style="list-style-type: none"> 1. Add new Room Rates/ Facilities 2. Modify Room Rates/ Facilities 3. Delete Room Rates/ Facilities
Alternative flows	<ol style="list-style-type: none"> 1. If the Room Rates/ Facilities code already exist, do not allow to add 2. If the Room Rates/ Facilities code is used in transactions do not allow deleting
Post- conditions	Display saved details
Actors	Company Management
Related Use Cases	

ID	U1.24
Name	Maintain Accommodation Type
Brief Description	Company Management defines Accommodation Type
Pre-conditions	Company Management has successfully logged into the system.
Basic flow of events	<ol style="list-style-type: none"> 1. Add new Accommodation Type 2. Modify an existing Accommodation Type 3. Delete an existing Accommodation Type
Alternative flows	<ol style="list-style-type: none"> 1. If the Accommodation Type code already exist, do not allow to add 2. If the Accommodation Type code is used in transactions do not allow deleting
Post- conditions	Display saved details
Actors	Company Management
Related Use Cases	

ID	U1.25
Name	Maintain Floors
Brief Description	Company Management defines hotel floors
Pre-conditions	Company Management has successfully logged into the system.
Basic flow of events	<ol style="list-style-type: none"> 1. Add new Floors 2. Modify/Delete existing floors
Alternative flows	<ol style="list-style-type: none"> 1. If the floor code already exist, do not allow to add 2. If the floor code is used in transactions do not allow deleting
Post- conditions	Display saved details
Actors	Company Management
Related Use Cases	

ID	U1.26
Name	Register Employee
Brief Description	HR Manager add employee details to the system
Pre-conditions	HR Manager has successfully logged into the system.

Basic flow of events	1. Add new Employee Details
Alternative flows	1. If the Employee No already exist, do not allow to add
Post- conditions	Successfully entered Employee Details
Actors	HR Manager
Related Use Cases	

ID	U1.27
Name	Update Employee Status
Brief Description	HR Manager updates employee status
Pre-conditions	<ul style="list-style-type: none"> 1. HR Manager has successfully logged into the system. 2. He has added successful employee details
Basic flow of events	Change employee status
Alternative flows	
Post- conditions	Display saved status
Actors	Company Management
Related Use Cases	

ID	U1.28
Name	Update Employee Profile
Brief Description	HR Manager add employee details to the system
Pre-conditions	<ul style="list-style-type: none"> 1. HR Manager has successfully logged into the system. 2. He has added successful employee details
Basic flow of events	<ul style="list-style-type: none"> 1. Modify Employee Details 2. Inactive Employee Details
Alternative flows	
Post- conditions	Successfully saved Employee Details
Actors	HR Manager
Related Use Cases	

3.3.3. Class Diagram

This diagram (**Figure 3.3**) describes the static overview of the system using Unified Modeling Language (UML) notation to model this static structure diagram that describes the structure of a system by showing the hotel management system's classes, their attributes, and the relationships between the existing entities.

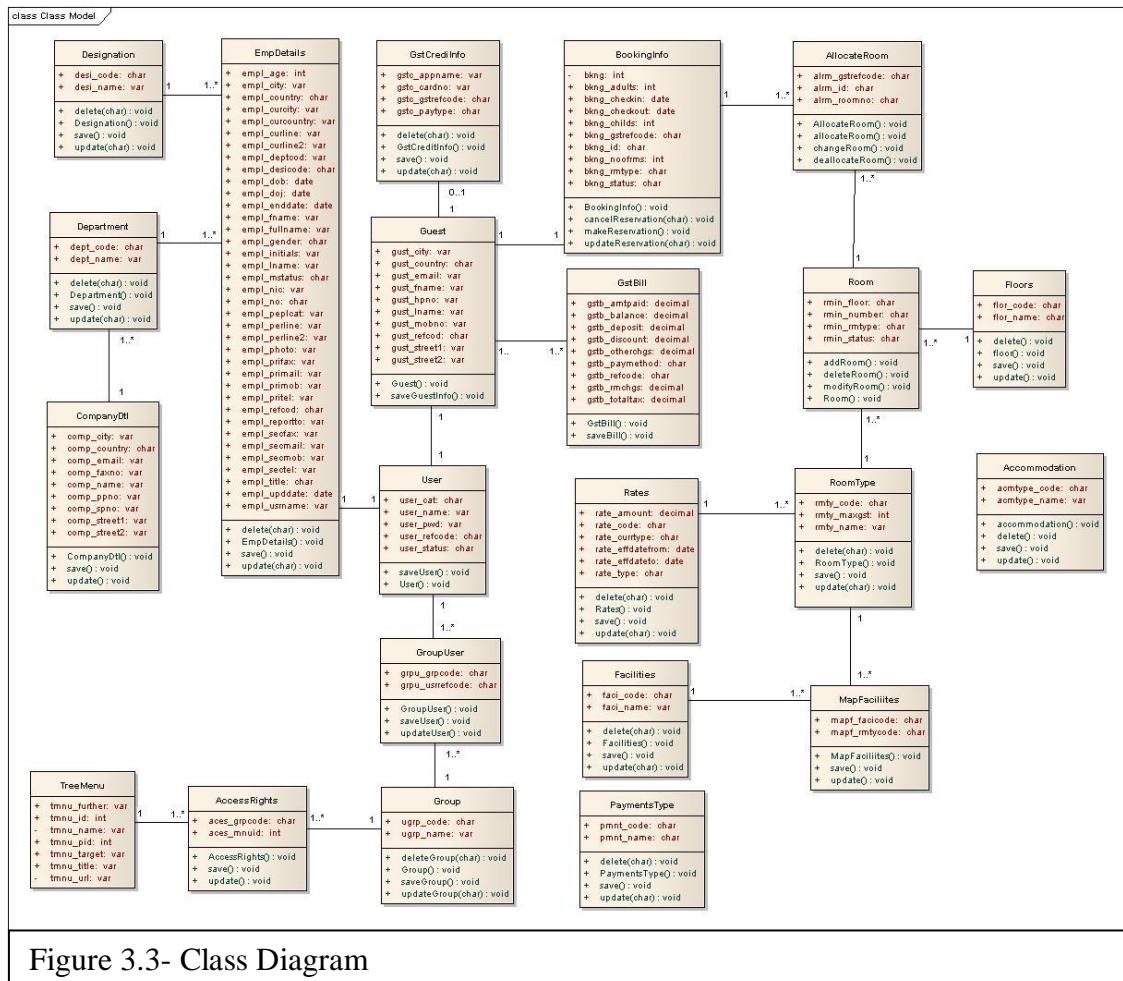


Figure 3.3- Class Diagram

3.3.4. Activity Diagram

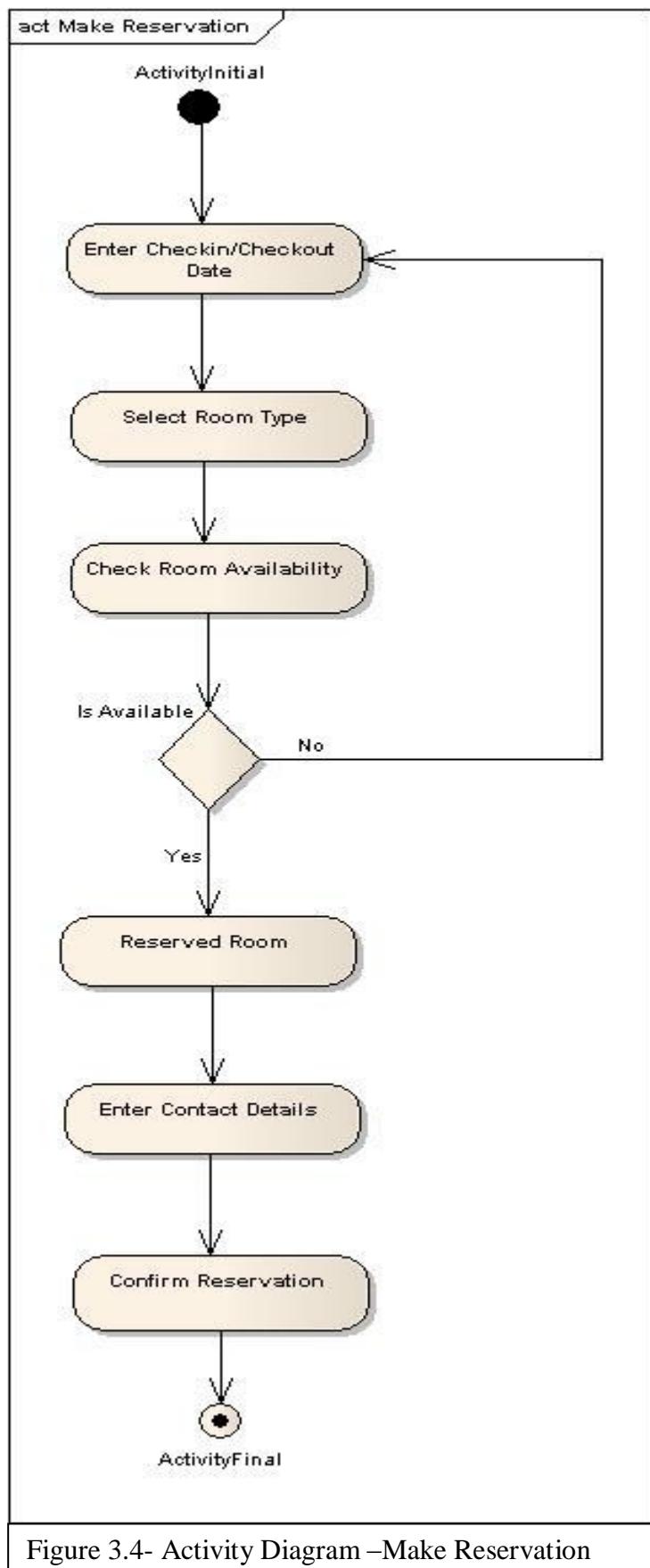
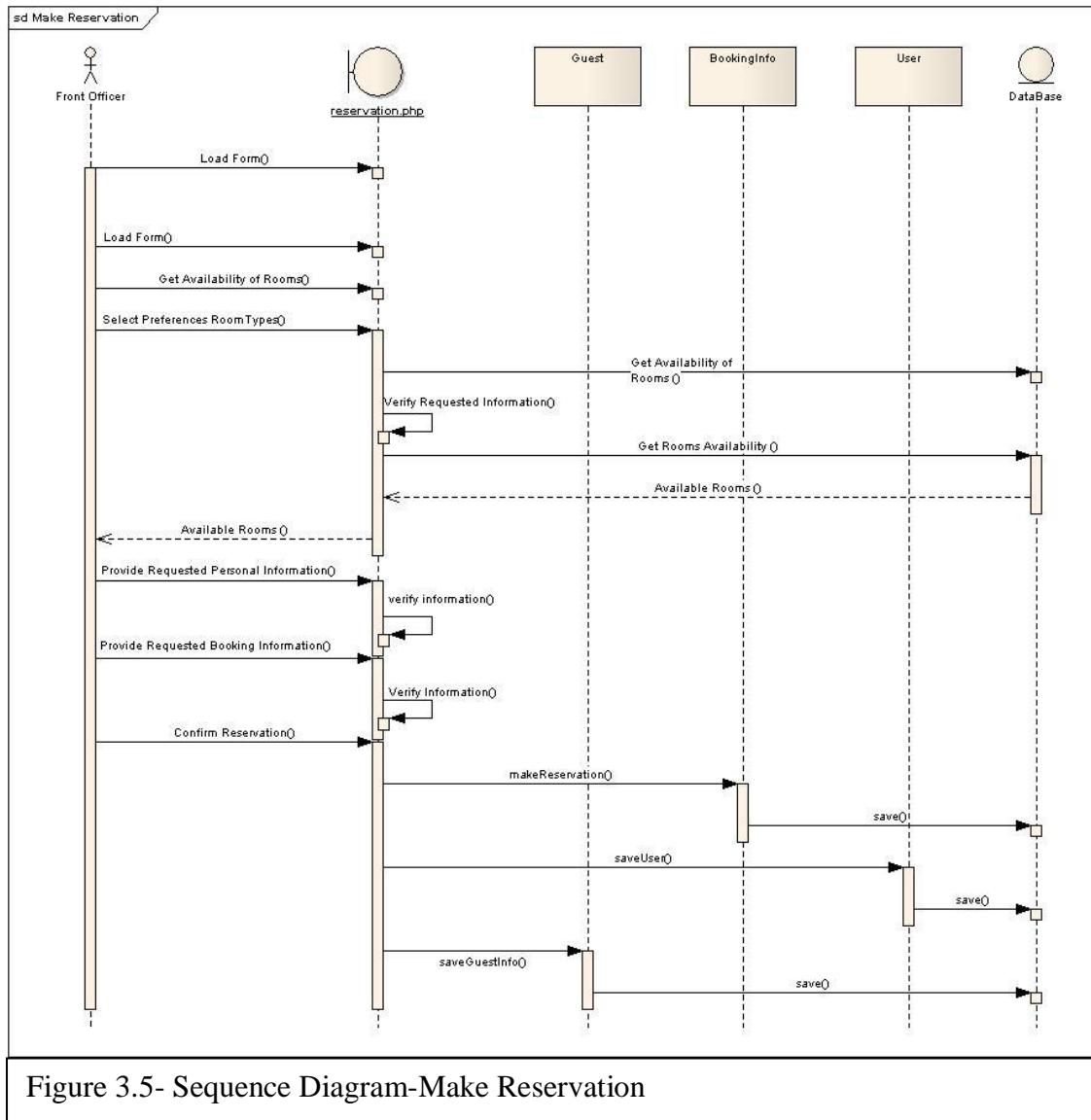


Figure 3.4- Activity Diagram –Make Reservation

The above activity diagram (**Figure 3.4**) describes process of the system which helps the system to complete make reservation functionality. This will be initiated by the customer and he will have a complete reservation record added to the system in successful completion.

3.3.5. Sequence Diagram

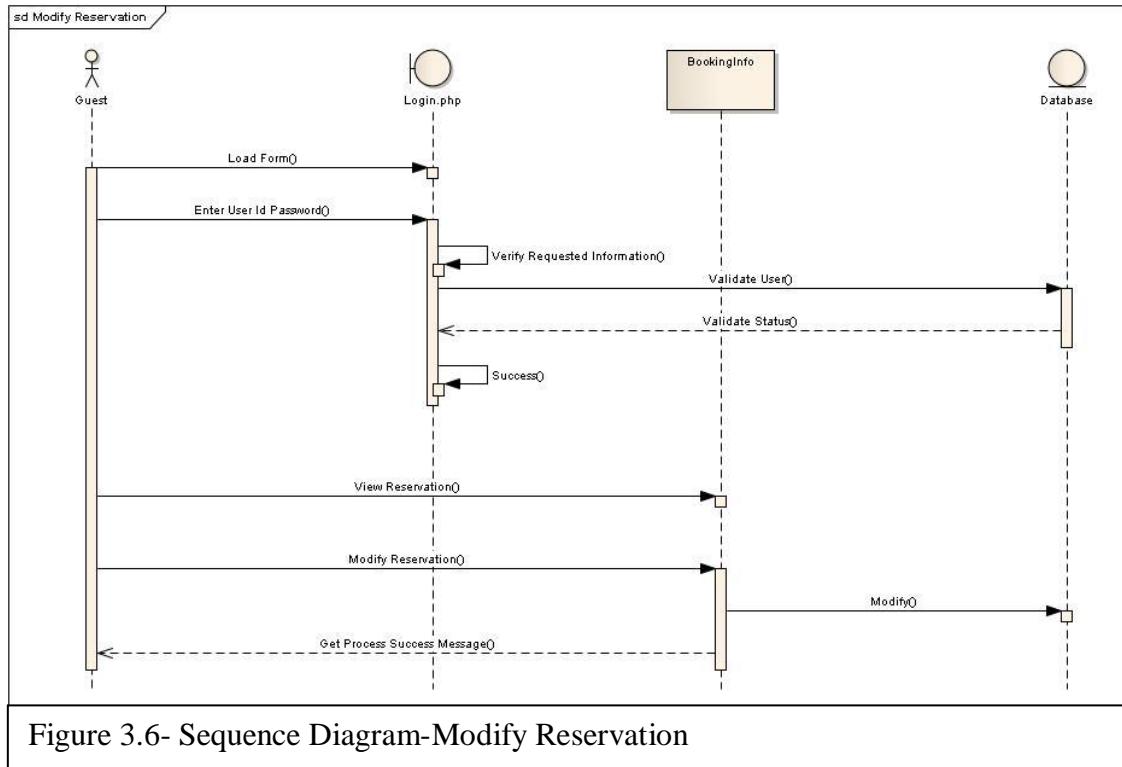
3.3.5.1 Make Reservation Related Sequence Diagram



The above diagram(**Figure 3.5**) describes the process of ‘make reservation’ functionality together with the interfaces, external entities and DB involved with the process of successful achievement of make reservation. Make reservation was a major concern of the client.

The diagram describes client viewing the room availability and selecting the prefferred facilities and make the reservation and how the database is update to reflect changes across other entity activities.

3.3.5.2 Modify Reservation Related Sequence Diagram



The above diagram (**Figure 3.6**) will describe how this web client interface works in conjunction with the system entities to achieve the modify reservation functionality of the system. This is initiated either by the customer or a front office when they required to update any details regarding an added reservation.

3.3.5.3 Cancel Reservation Related Sequence Diagram

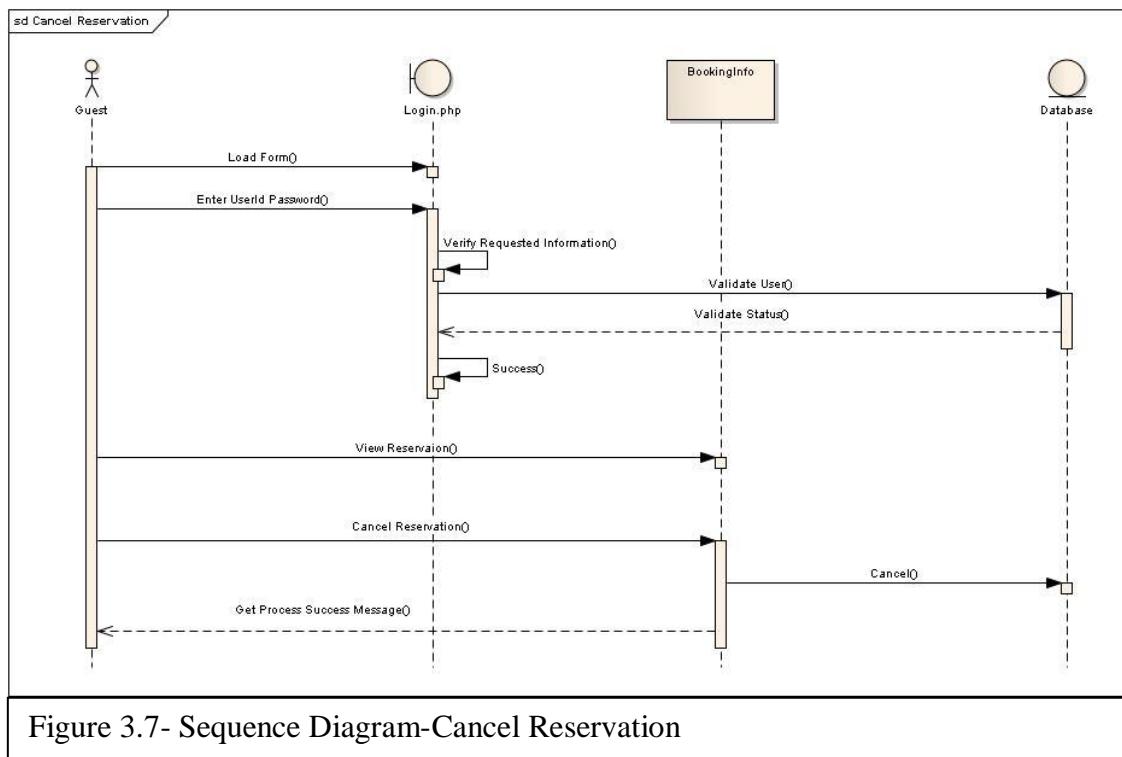


Figure 3.7- Sequence Diagram-Cancel Reservation

The above diagram (**Figure 3.7**) describes ‘cancel Reservation’ process of the system and how the booking info entity ceases to exist after it was deleted by the customer after the reaervation is cancelled by the customer or front office when required.

3.3.6. Entity Relationship Diagram

This system was designed using Object oriented methodology following diagram is an abstract and conceptual representation of the database (3.8).The logical and physical representation of system entities and their relationships was designed using an Entity Relationship diagram (Figure 3.8) for the project.

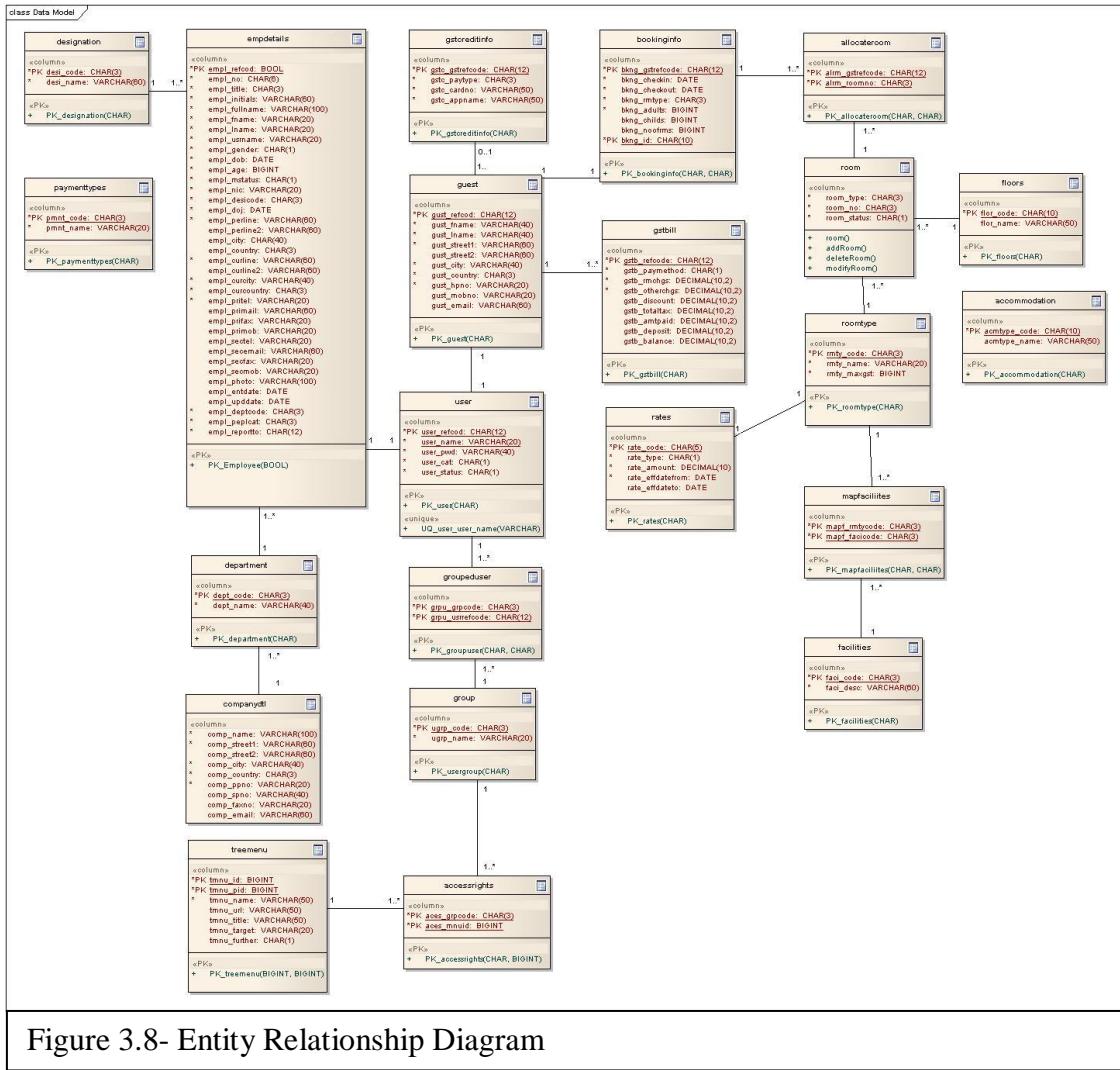


Figure 3.8- Entity Relationship Diagram

3.4. User Interfaces

The user interface defines how users interact with a system. Not only has an interface to interact it presented information in an organized manner to user. User interface designing should provided by the best way for users to interact with system. Good interface design leads to user satisfaction.

3.4.1. Evaluation Criteria of a Good User Interface

Following are the four usability metrics to measure quality of a user interface.

- Analytical metrics –whether all the needed information appears on the screen
- Performance metrics – time used to perform a task, system robustness
- Cognitive workload metrics – the mental effort required to use the system.

- User satisfaction metrics – how helpful the system is and how easy it is to learn.

3.4.2. System Navigation

The system can be loaded from the link provided in the web site of the client organization. After login in to the system, the authorized users would be able to navigate among the 8 modules and the user interfaces of them.

3.4.3. Interface Standard

All the module user interfaces have a common format with link to load each screen. Style (font-family, color, font sizes, and background-color) of the screen are standardized using Cascading Style Sheets (CSS). Structure of the interface was displayed using HTML. JavaScript was used as the client-side scripting for handling interface behavior and validation. PHP scripting function used to include common template of each screen. 1024 x 768 pixels resolution was used as a standard in interface designing. To indicate mandatory fields of the system was displayed with a ‘Red Asterisk’ mark.

3.4.4. List of Input Screens

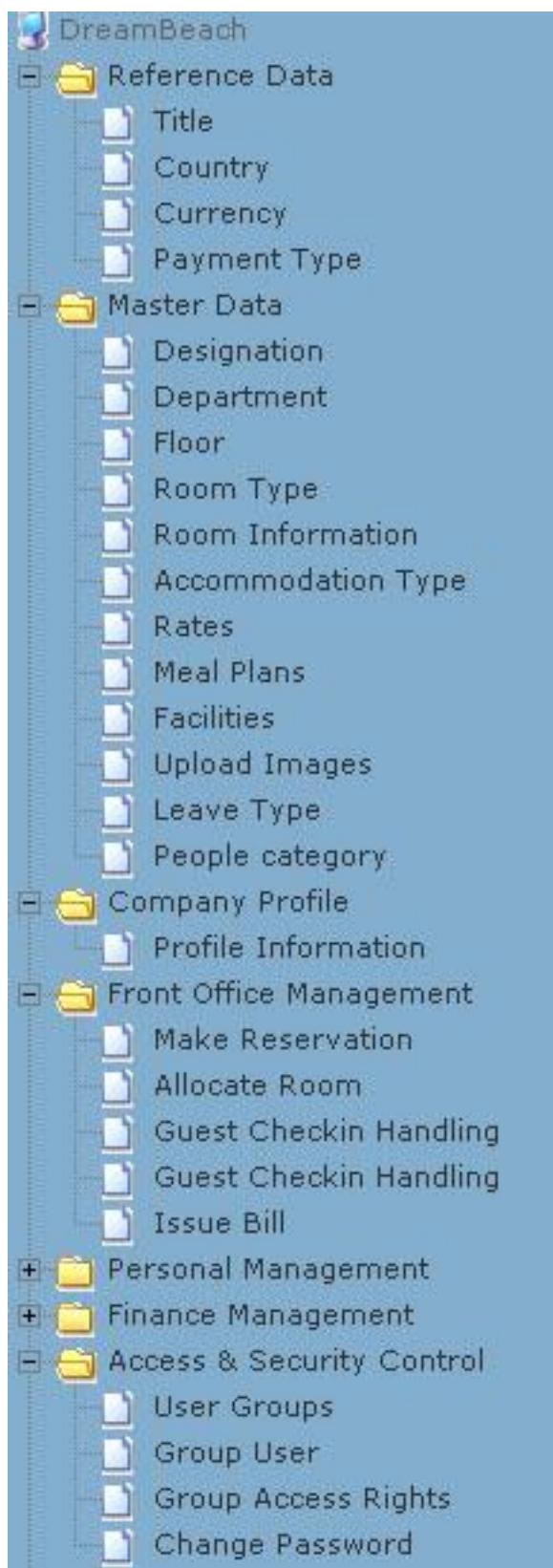


Figure 3.9 List of input screens

List of input screens of the developed system displayed in **Figure 3.9**

3.4.5. List of Output Screens



Figure 3.10 List of output screens

List of output screens of the developed system displayed in **Figure 3.10**

3.4.6. Sample User Interfaces

3.2.6.1 Home Page of the Web Site



Figure 3.11 Home page of the system

Home page (**Figure 3.11**) of the developed system provides users an interactive interface which has information listed under following categories as ‘Home’, ‘About us’, ‘Reservation’, ‘Facilities’, ‘Contact’ us and ‘Guest’.

Home page gives the overview description of the hotel through an attractive interface like displaying the welcome message, highlighting the facts like why customers should select them and displaying special offers given on time to time.

About us will display the rates and facilities provided based on selected country or region.

Reservation will provide the facility of selecting and reserving the meal plan, room types, duration of stay, etc... to the customer.

3.2.6.2 Reservation Page of the Web Site

Dream BEACH

We serve you the best

Home | **About Us** | **Reservation** | **Facilities** | **Contact Us** | **Guest Login**

Select Your Stay Days

Check-in Date: *
Check-out Date: *

Select Your Room Type

Room Type	Rate	Rooms
Effective Date From 2010-06-13 To 2010-12-13		
Double	40.00\$	<input type="button" value="Check Availability"/> 0
Effective Date From 2010-06-13 To 2010-12-13		
Single	30.00\$	<input type="button" value="Check Availability"/> 0
Effective Date From 2010-06-13 To 2010-12-13		
Triple	50.00\$	<input type="button" value="Check Availability"/> 0

Enter Personal Information

Title: *
First Name: *
Last Name: *
Gender: Male Female *
Phone Number: *
NIC No: *
Street: *
City: *
Country: China
E-mail:

Security Information

User Name: *
Password: *
Confirm Password: *

Enhance Your Stay

Days: Adults: Childs:

Continue | Reset

2010@All Rights Reserved by Dream Beach Resort

Figure 3.10 Reservation page of the system

Reservation page (**Figure 3.12**) of the developed system provides users to do their reservation. It offers the end user maximum comfort in his ability to select arrival and departure date; select his room type with room rates from an orderly set of choices. Rooms can be visualized and total prices as well as availability are displayed in real time.

3.2.6.3 Login Screen

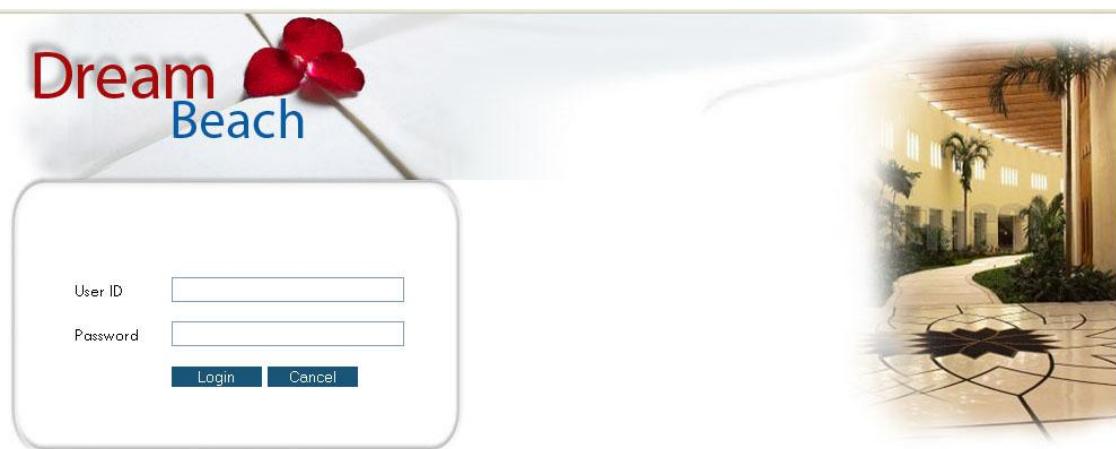


Figure 3.11 Login Screen

When system was loaded from the company website, the user login screen (**Figure 3.13**) will be displayed. As only the employees of the company would be the users of the current system. Only the authorized users will be able to log in to the system. Unauthorized login attempts would get an error message. (**Figure 3.14**)

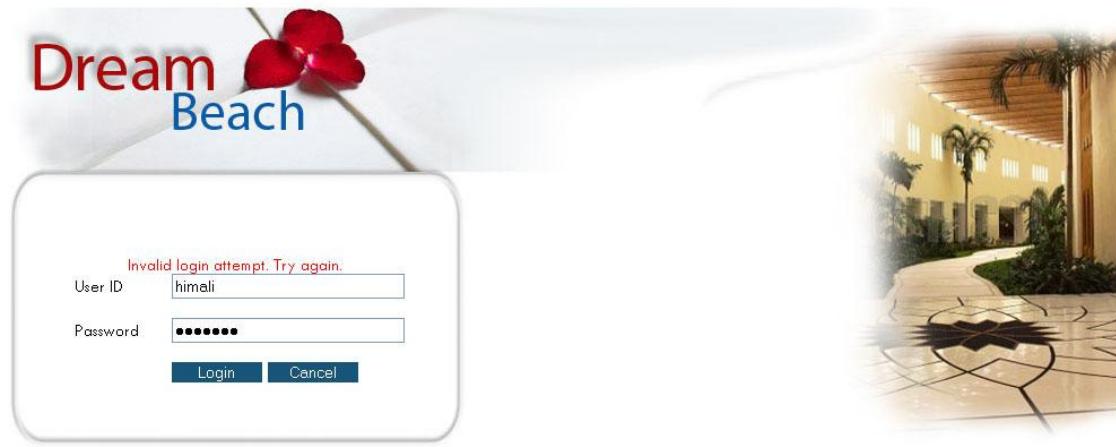


Figure 3.14 Unauthorized login Attempt

3.2.6.4 Main Interface of the System

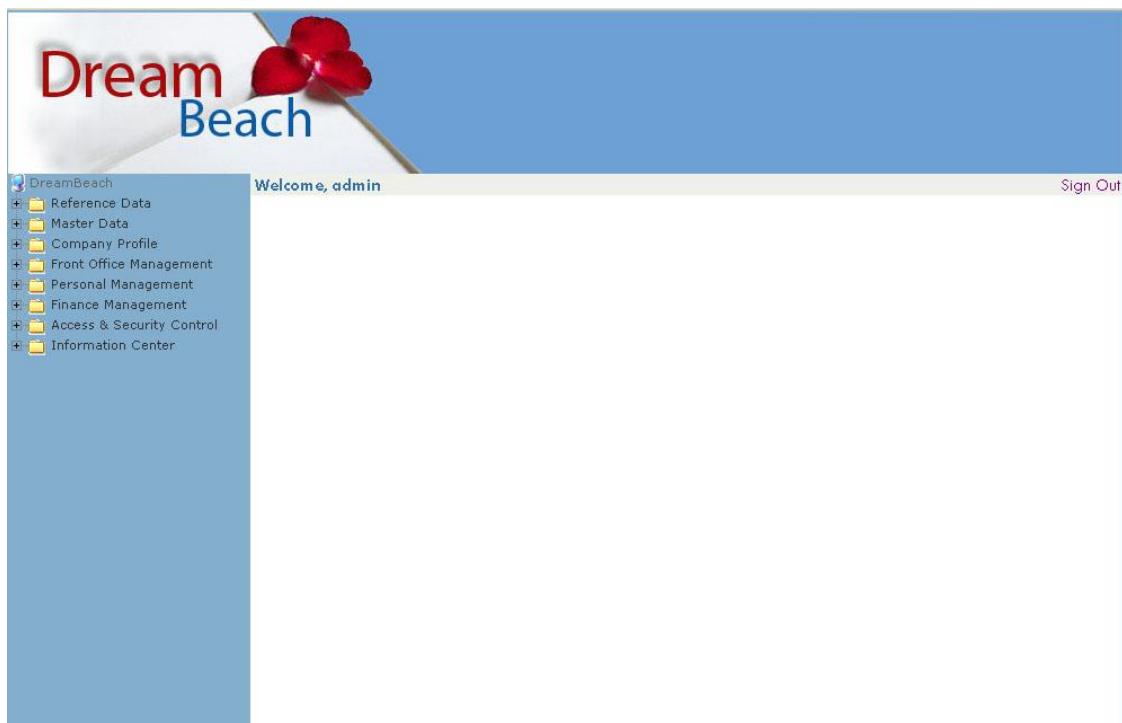


Figure 3.15 Main Interface of the System

After successfully logged in to the system, the user would see the module structure of the system (**Figure 3.15**).

- Reference Data Module - The globally available data such as countries, person titles can be said as the reference data. They are not frequently update.
- Master Data Module - Module to define and maintain the master data commonly used in other modules.
- Company Profile- Module to update company profile
- Front Office Management- Module to handle reservation, check-in, check-out, room allocation and deallocation
- Personal Management - Module to maintain the details of human resources.
- Finance Management – Module to analysis daily income
- Access & Security Control - Module to maintain the system security.
- Information Center - Module to view the system reports and information screens.

3.2.6.5 Country Detail Maintenance (Reference Data Module)

The screenshot shows the 'Define Country' screen within the Dream Beach application. The left sidebar contains a navigation tree with nodes such as Reference Data (Title, Country, Currency, Payment Type), Master Data, Company Profile, Front Office Management, Personal Management, Finance Management, Access & Security Control, and Information Center. The main content area has a title 'Welcome, admin' and a sub-title 'Define Country'. A search bar is present with a dropdown menu labeled '--Select--' and a search button. Below it is a data grid table with two columns: 'Country Code' and 'Country Name'. The table lists 11 records from 1 to 10, including Australia, China, Canada, England, India, Japan, Korea, Moldives, Pakistan, and Sri Lanka. At the bottom of the grid are buttons for 'Total Records: 11' and 'Displaying: 1 - 10'. Below the grid is a form with fields for 'Country Code' and 'Country Name', both marked with a red asterisk. At the very bottom are buttons for 'Add', 'Modify', 'Delete', 'Save', 'Clear', and 'Exit'.

	Country Code	Country Name
1	AUS	Australia
2	CHN	China
3	CND	Canada
4	END	England
5	IND	India
6	JPN	Japan
7	KRN	Korea
8	MOL	Moldives
9	PKS	Pakistan
10	SRI	Sri Lanka

Figure 3.16 Country Details Maintenance

New Country can be added to the system using this screen (**Figure 3.16**) by adding Country Code and Country Name. The particular user logged in to the system and adds the new record using provided screen. Also, the details of existing country can be modified and deleted using this interface.

All the reference data defining and maintaining screens contain the same interface formats with similar button behaviors.

Button Actions:

Add- Clears and enables the input fields to add a new record. Data grid will be disabled.

Modify- Enables the modifiable input fields and displays the details of the selected record in them. Data grid will be disabled.

Delete – Deletes or inactivates the selected record.

Save – Save the added or modified record.

Clear – Clear the input fields and enables the data grid.

Close – Close the currently viewing user interface.

3.2.6.6 Room Type Maintenance (Master Data Module)

Room Type Code	Room Type Name	Maximum Guest
1 DBL	Double	2
2 SIN	Single	1
3 TPL	Triple	3

Figure 3.17 Room Types Maintenance

This user interface (**Figure 3.17**) provides the facility to system user to define room types using Room Type Code, Room Type Name, and Maximum guest per room. . Also, the details of existing Room Type can be modified and deleted using this interface.

3.2.6.7 Company Profile (Company Profile)

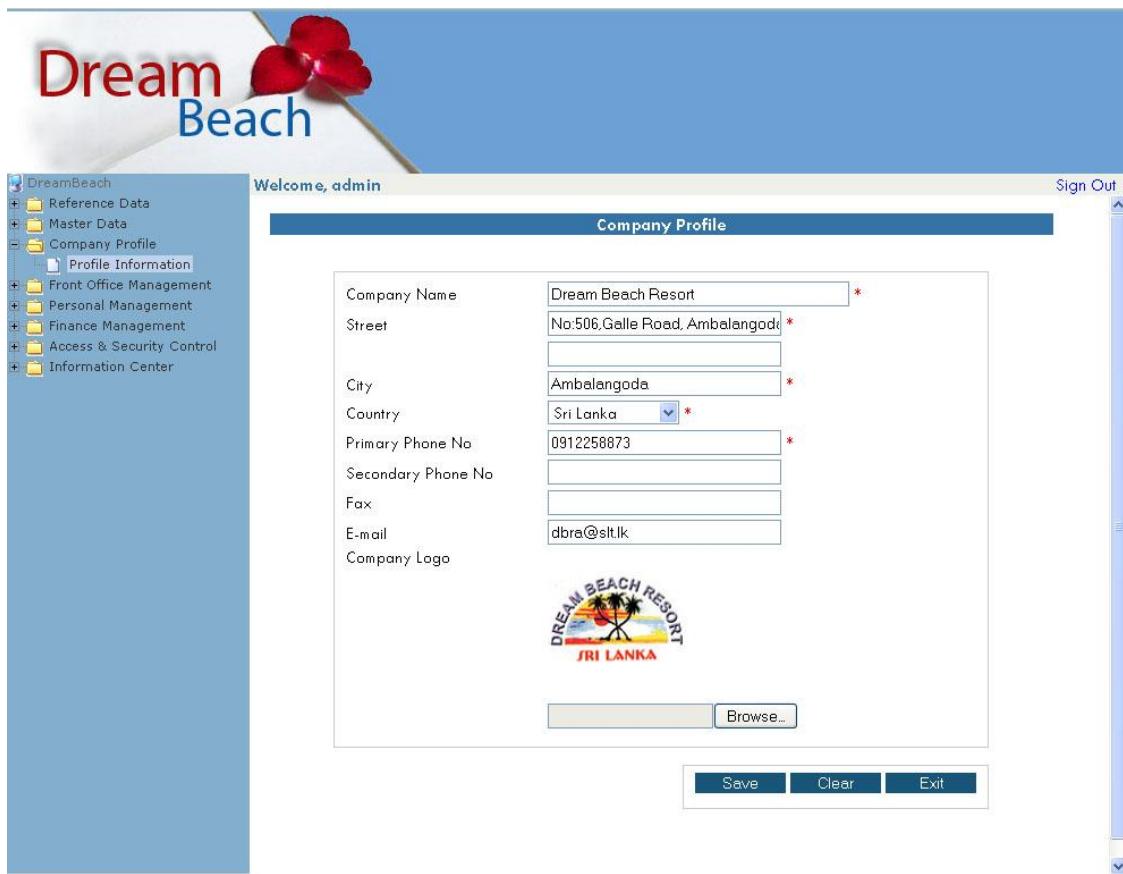


Figure 3.18 Company Profile

This user interface (**Figure 3.18**) provides the facility to system user maintain their company profile.

3.2.6.8 Access Rights Interface (Access and Security Control Module)

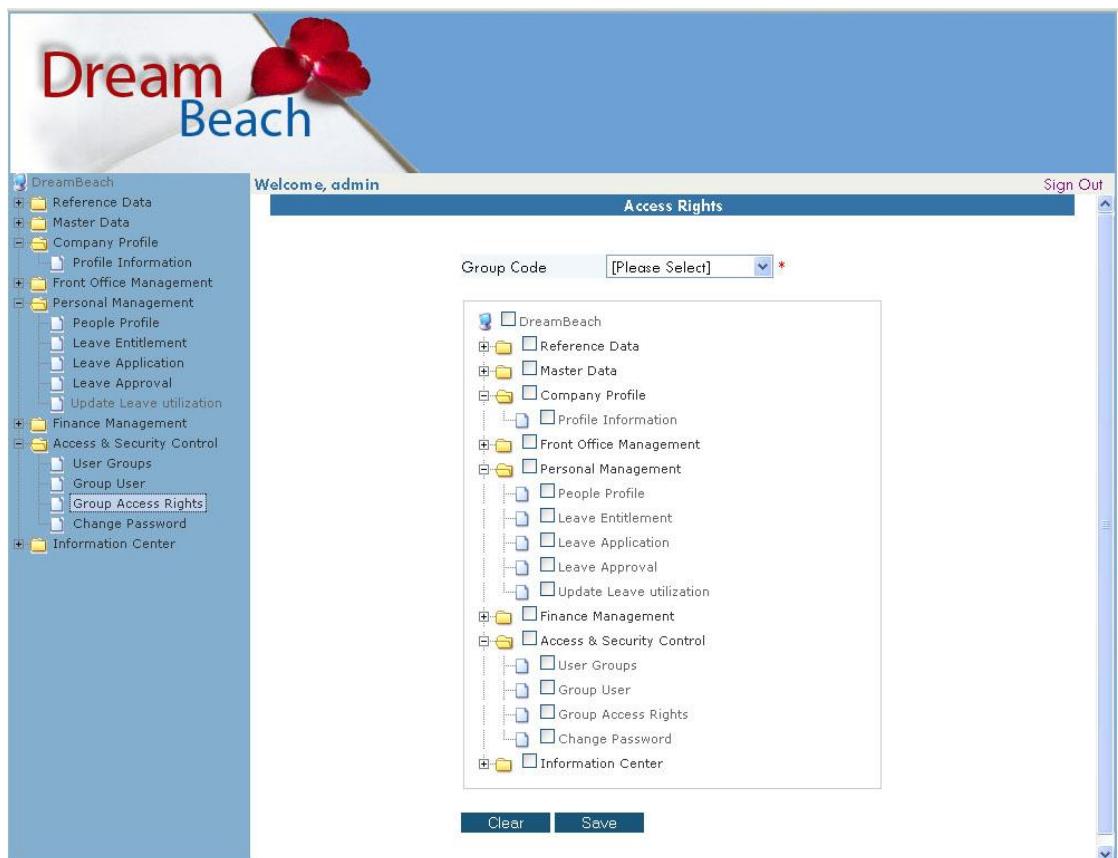


Figure 3.19 Access Rights Interface

Access rights can be granted to the security user groups defined by using the 'Define User Groups' screen (**Figure C.4**). Already granted permissions can be revoked by removing the assigned screens.

Chapter 4 – Implementation

This chapter contains construction phase of the RUP. Construction means to build the project. During the construction phase all remaining components and application features are developed and integrated into the product, and all features are thoroughly tested.

Objectives of this phase:

- Minimize development costs by optimizing resources and avoiding Online room reservation
- Achieve useful version
- Achieve adequate quality
- Complete the analysis, design, development and testing of all required functionality
- Income analysis

The outcome of the construction phase is a product ready to put in the hands of the end users.

It consists of

- The Software integrated on the adequate platform
- The user manual

4.1 Implementation Environment

4.1.1 Server side implementation

An operating system should be installed on a computer which is considered as the server. The system was developed and tested on MS windows XP Professional.

Apache HTTP server 2.0.x should be installed on the computer to perform as the web server. PHP 5.x should be configured with the installed Apache web server, and MySQL 5.x database server also should be configured. The installations and configurations of these softwares are explained in Appendix A.

The server computer of the system should have an email server to send email. At development, cMail Server was used as the email server.

4.1.2 Client side implementation

The client should have an internet connected computer with a web browser to work with the system

4.2 Developments Tools

- **Macromedia Dreamweaver 8**

The complete interface designing and coding was done using Macromedia Dreamweaver. The usability of the software was high and web pages could be designed and coded easily. HTML, JavaScript, CSS and PHP scripting were supported in Dreamweaver with identifiable code format settings.

- **MySQL Administrator 5.0**

MySQL Administrator which is under GPL license was used in system development for database related activities. It is a GUI client tool which is helpful in exploring the database structures and performing data related operations.

- **Adobe Photoshop**

Colorful and meaningful images would make websites and software more attractive for the users and customers. Adobe Photoshop was used for creating and editing the images, icons displayed in the software system and the web site.

- **cMailServer 4.3.1**

CMailServer 4.3.1 was used as the email server. It is a free email server for Windows and support for IMAP, POP3 and SMTP email protocols. Open source and under GNU-GPL license.

4.3 Hardware and software Requirements

Server side hardware and software requirements for this system described in the Table 4.1

4.3.1 Server side Hardware and Software Requirements

Hardware Requirements	<ul style="list-style-type: none">• 3GB or more hard disk capacity• 512MB or above RAM• Pentium IV Computer
Software Requirements	<ul style="list-style-type: none">• PHP version 5.0.x• Apache ,MySQL & PHP supported OS (Linux / Windows)• MySQL 5.x database server• Apache Web Server 2.0.x

	<ul style="list-style-type: none"> • An Email Server (Eg: cMailServer for Windows)
--	---

Table 4.1 server side hardware and software requirements

4.3.2 Client side Hardware and Software Requirements

Client side hardware and software requirements for this system described in the Table 4.2

Hardware Requirements	<ul style="list-style-type: none"> • Pentium compatible computer • 256MB RAM • SVGA monitor (1024 x 768 resolution)
Software Requirements	<ul style="list-style-type: none"> • A Windows 2000 or later OS or Linux OS • JavaScript enabled web browser (Eg: IE 6+ or Mozilla Firefox 2+)

Table 4.2 client side hardware and software requirements

4.4 Code and Module Structure

Following is the structure (**Figure 4.1**) of files and folders of the system. The main folder ‘dreambeah’ is located inside the ‘htdocs’ folder in Apache directory, as it is the location targeted as ‘localhost’ of the server machine.

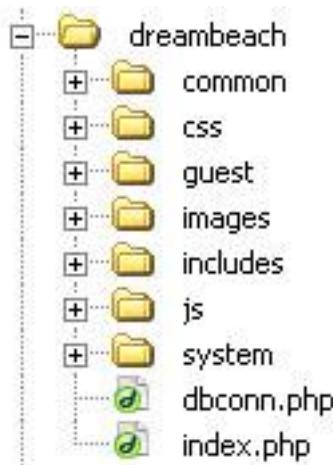


Figure 4.1 Structure of the files and folders of Dreambeach

The **common** folder contains:

- All the common functions such as ‘getCountry()’, ‘getTitle()’ etc.
- Grid handling ajax functions such as `handleGridAjaxRequests ()` etc.

The **css** folder contains:

- All the cascading styles used in the system

The **guest** folder contains files related to the guest panel of the system.

- The classes of the guest panel
- File contains ajax functions used in the each interface
- Client side and web tier coding with HTML ,Java Script and PHP

The **images** folder contains:

- All the images used in the system

The **includes** folder contains:

- Common templates used in the system such as '**header.php**', '**footer.php**' etc.

The **js** folder contains:

- Validation functions of the system.

The **system** folder contains files related to the each module of the system.

- The classes of the each module
- File contains ajax functions used in the each interface
- Client side and web tier coding with HTML ,Java Script and PHP

dbconn.php – data base handling class

Index.php - The first page loaded when the system is called through the web site of the client organization.

4.5 Sample Codes

Some of the code segments belonging to Master data Module and included in this section. Other important codes are included in Appendix E.

4.5.1 Web Tier with PHP scripting (add_roomtype.php)

```
<?php
if($_SERVER['REQUEST_METHOD'] == "POST"){
    $roomtypCode = trim($_REQUEST["roomtypCode"]);
    $roomtypName = trim($_REQUEST["roomtypName"]);
    $maxGst      = trim($_REQUEST["maxGst"]);
    $hidProcess  = trim($_REQUEST["hidProcess"]);
    $err_no =0; //if error is occurred check where it is
//Check empty fields
    if($roomtypCode==""){
        $err_no++;
        $error_msg1=validateMandatory($label1);
    }
    if($roomtypName==""){
        $err_no++;
        $error_msg2=validateMandatory($label2);
    }
    if($maxGst==""){
        $err_no++;
        $error_msg3=validateMandatory($label3);
    }
    if($err_no==0){
        $arrayVal=array($roomtypCode,$roomtypName,$maxGst);
        $rmt=new Roomtype($arrayVal);
//Call add function
        if($hidProcess=="add"){
            $msg=$rmt->save();
        }
//Call update function
        if($hidProcess=="alter"){
            $msg=$rmt->update();
        }
//Call remove function
        if($hidProcess=="remove"){
            $msg=$rmt->remove();
        }
    }
}
?>
```

4.5.2 Class with attributes and functions

```

<?php
class Roomtype{
    var $roomtypCode;
    var $roomtypName;
    var $maxGst;
    function Roomtype($vals){
        $this->roomtypCode=$vals[0];
        $this->roomtypName=ucwords($vals[1]);
        $this->maxGst=$vals[2];
    }
    //function to Add Room Type details
    function save(){
        $db_err_count=0;
        //Check whether the Room Type code is already exist.
        $select_query="SELECT * from roomtype where rmty_code='".$this->roomtypCode."'";
        $result_query=mysql_query($select_query);

        if(mysql_num_rows($result_query)>0){
            return "<font color='red'><b>$this->roomtypCode</b> Code is Already Exist</font>";
        }
        else{
            $insert_query="insert into roomtype(rmty_code,rmty_name,rmty_maxgst)values('".$this->roomtypCode."','".$this->roomtypName."','".$this->maxGst")";
            if(!@mysql_query($insert_query)){
                mysql_query("rollback");
                if(trigger_error){$err ="01";}
            }
            if(!mysql_query("commit")){
                mysql_query("rollback");
                trigger_error("An Error Occured...", E_USER_ERROR);
            }
            else{
                return "Record Added Successfully";
            }
        }
    }
    //function to Update Room Type details.
    function update(){
        $date= date("Y-m-d");
        $db_err_count=0;
        $update_query="UPDATE roomtype set rmty_name='".$this->roomtypName',rmty_maxgst='".$this->maxGst'
where rmty_code='".$this->roomtypCode."'";
        if(!@mysql_query($update_query)){
            mysql_query("rollback");
            if(trigger_error){$err ="01";}
        }
        if(!mysql_query("commit")){
            mysql_query("rollback");
            trigger_error("An Error Occured...", E_USER_ERROR);
        }
        else{
            return "Record Updated Successfully.";
        }
    }
    // Function to Delete Room Type details.
    function remove(){
        $delete_query="DELETE from roomtype where rmty_code='".$this->roomtypCode."'";
        if(!@mysql_query($delete_query)){
            mysql_query("rollback");
            if(trigger_error){$err ="01";}
        }
        if(!mysql_query("commit")){
            mysql_query("rollback");
            trigger_error("An Error Occured...", E_USER_ERROR);
        }
        else{
            return "Record Deleted Successfully.";
        }
    }
}
?>

```

Chapter 5 – Evaluation

This chapter covers the testing and evaluation of the system

5.1 System Testing

Testing is one of the most important stages in the software process. The system-testing phase was carried out in order to ensure that the system conforms to its specifications and meets the needs of the user. Therefore, software testing was done to validate and verify the software system. The main objective of this phase is to ensure that the system functions according to its requirements.

5.1.1. Testing Phases

Unit Testing - programs were tested individually to ensure that their functioning correctly.

Module Testing- collections of programs were tested to elicit the hidden defects which were not identified during unit testing.

Integration Testing - Tested modules were integrated and tested concerning on their dependencies with each other.

System Testing - Integrated modules were tested concentrating on validating the system that it meets its functional and non-functional requirements.

User Acceptance Testing - The system was finally tested by the end users using the actual data.

5.1.2. Test Plan

The major test plan techniques were used to designing test cases.

Black Box Testing

Actual users were also included in the black box testing phase. Black box testing concerns only the functional specification (external view) of the system without any knowledge of the internal structure of the software

White Box Testing

White box test cases require thorough knowledge of the internal structure of the software, and therefore, the white box testing is called structural testing. The test is

accurate only if the tester knows what the program is supposed to do and how it is done.

5.1.3. Test Cases

All the scenarios which actual user goes through in accessing system were tested. Real users were also included in the evaluation of these test cases (**Table 5.1**) in order to get the suitable conclusion.

Test Case Id	1			
Tested Component	Login Page			
Test Case Description				
No	Test Case	Expected Output	Actual Output	Test Status
1	User enter valid user id & password	Load the modules tree structure of system according the user id	Load the modules tree structure of system according the user id	Ok
2	User enter valid user id & invalid password	Display an error message	Display an error message	Ok
3	User enter invalid user id & valid password	Display an error message	Display an error message	Ok
4	User enter valid user id & password with blank	Display an error message	Display an error message	Ok
5	User enter valid password & user id with blank	Display an error message	Display an error message	Ok
6	User click [Cancel] button	Clear the value of text feild	Clear the value of text feild	Ok

Table 5.1 Test Case Descriptions –Login Page

Following test cases (**Table 5.2**) were used to test the system against the maintain room type function handled by the admin user.

Test Case Id	2			
Tested Component	Room Type			
Test Case Description				
No	Test Case	Expected Output	Actual Output	Test Status
1	Click [Add] button	Disable the grid. Enable input fields. [Room Type code] ,[Room Type Name] and [Maximum Guest]	Grid disabled. Enabled input fields.	Ok
2	Enter [!,@,#,\$,%,&,*] characters for [Room Type code] input field.	Disable key press for [!,@,#,\$,%,&,*] characters	Disabled key press for [!,@,#,\$,%,&,*] characters	Ok
3	Enter [A-Z, a-z & 0-9] characters for [Room Type code] input field.	Enable key press for [A-Z , a-z & 0-9] characters	Enabled key press for [A-Z , a-z & 0-9] characters	Ok
4	Enter [A-Z	Disable key press for	Disable key press for	Ok

	and!,@,#,\$,%,^,&,*] characters for [Maximum Guest] input field.	[A-Z and !,@,#,\$,%,^,&,*] characters	[A-Z and !,@,#,\$,%,^,&,*] characters	
5	Enter mandatory fields value & click [Save] button	Display save status message	“Record Added Successfully” message displayed.	Ok
6	Click [Save] button without entering mandatory fields value	Display an error message	Error message displayed.	Ok
7	Save already existing [Room Type Code] field value	Display an error message	“[Room Type Code] Code is Already Exist” message displayed.	Ok
8	Select existing value from grid	Display value in input fields	Value displayed in input fields.	Ok
9	Select existing value from grid & click [Modify] button	Enable modifiable input fields of the selected record.	Enabled modifiable input fields of the selected record.	
10	Click [Save] button to save modified data	Display save status message	“Record updated Successfully” message displayed.	Ok
11	Select existing value from grid & click [Delete] button	Disable all input fields of the selected record & display delete confirmation message to delete the selected record. If confirmed, delete the record. Else do not delete the record.	Confirmation message displayed. If click [Yes] record delete. if click [No] record remained.	Ok
12	Click [Clear] button	Clear the input fields	input fields cleared	
13	Click [Exit] button	Display conformation message box to exist from the form. if confirm exist from form. Else does not exist.	Confirmation message displayed. If click [Yes] exist from current form. If click [No] form does not exist.	

Table 5.2 Test Case Descriptions – Room Type

Test cases given below (**Table 5.3**) will look after the scenarios in the add room rates function of the system.

Test Case Id	3			
Tested Component	Rates			
Test Case Description				
No	Test Case	Expected Output	Actual Output	Test Status
1	Click [Add] button	Disable the grid. Enable input fields.	Grid disabled. Enabled input fields.	Ok
2	Enter [A-Z ,a-z, !,@,#,\$,%,^,&,*] for [Amount] input field	Disable key press for [A-Z ,a-z, !,@,#,\$,%,^,&,*]	Disabled key press for [A-Z ,a-z, !,@,#,\$,%,^,&,*]	Ok
3	Enter [0-9] for [Amount] input field	Enable key press for [0-9]	Enabled key press for [0-9]	Ok

4	Enter mandatory fields value & click [Save] button	Display save status message	“Record Added Successfully” message displayed.	Ok
5	Click [Save] button without entering mandatory fields value	Display an error message	Error message displayed.	Ok
6	Save already existing record	Display an error message	“Record Failed.” message displayed.	Ok
7	Select existing value from grid	Display value in input fields	Value displayed in input fields.	Ok
8	Select existing value from grid & click [Modify] button	Enable modifiable input fields of the selected record.	Enabled modifiable input fields of the selected record.	
9	Click [Save] button to save modified data	Display save status message	“Record updated Successfully” message displayed.	Ok
10	Select existing value from grid & click [Delete] button	Disable all input fields of the selected record & display delete confirmation message to delete the selected record. If confirmed, delete the record. Else do not delete the record.	Confirmation message displayed. If click [Yes] record delete. if click [No] record remained.	Ok
11	Click [Clear] button	Clear the input fields	input fields cleared	
12	Click [Clear] button to delete a record with existing related data	Display an error message	Displayed an error message	
13	Click [Exit] button	Display conformation message box to exist from the form. if confirm exist from form. Else does not exist.	Confirmation message displayed. If click [Yes] exist from current form. If click [No] form does not exist.	

Table 5.3 Test Case Descriptions – Room Pricing

The following test cases (**Table 5.4**) were executed on the system inorder to test the system against view employee profile function handled by the HR Manger.

Test Case Id	4			
Tested Component	People Profile			
Test Case Description				
No	Test Case	Expected Output	Actual Output	Test Status
1	Click [Add] button	Disable the grid. Enable input fields.	Grid disabled. Enabled input fields.	Ok
2	Enter [A-Z ,a-z, !,@,#,\$,%,&,*] for [Telephone][Mobile] input field	Disable key press for [A-Z ,a-z, !,@,#,\$,%,&,*]	Disabled key press for [A-Z ,a-z, !,@,#,\$,%,&,*]	Ok
2	Enter [0-9] for	Enable key press for	Enabled key press for	Ok

	[Telephone][Mobile] input field	[0-9]	[0-9]	
3	Enter invalid email address	Display and error message	Display and error message	
5	Enter already exist value for [User Name] field	Display an error message.	"[User Name] Already exist" message displayed.	Ok
6	Enter value for [User Name] field, which not exist.	Display availability message	"[User Name] Available" message displayed.	Ok
7	Enter mandatory fields value & click [Save] button	Display save status message	"Record Added Successfully" message displayed.	Ok
8	Click [Save] button with invalid file format for [Image Path]	Display an error message	Error message displayed.	Ok
9	Click [Save] button without entering mandatory fields value	Display an error message	Error message displayed.	Ok
10	Save already existing record	Display an error message	"Record Added Failed." message displayed.	Ok
11	Select existing value from grid	Display value in input fields	Value displayed in input fields.	Ok
12	Select existing value from grid & click [Modify] button	Enable modifiable input fields of the selected record.	Enabled modifiable input fields of the selected record.	Ok
13	Click [Save] button to save modified data	Display save status message	"Record updated Successfully" message displayed.	Ok
14	Select existing value from grid & click [Delete] button	Disable all input fields of the selected record & display delete confirmation message to delete the selected record. If confirmed, delete the record. Else do not delete the record.	Confirmation message displayed. If click [Yes] record delete. if click [No] record remained.	Ok
15	Click [Clear] button	Clear the input fields	input fields cleared	
16	Click [Clear] button to delete a record with existing related data	Display an error message	Displayed an error message	
17	Click [Exit] button	Display conformation message box to exist from the form. if confirm exist from form. Else does not exist.	Confirmation message displayed. If click [Yes] exist from current form. If click [No] form does not exist.	

Table 5.4 Test Case Descriptions – Employee Details

The following test cases (**Table 5.5**) were run on the system to test the user group creation function of the system against all possible scenarios occur in the real user environment.

Test Case Id	5			
Tested Component	User Group			
Test Case Description				
No	Test Case	Expected Output	Actual Output	Test Status
1	Click [Add] button	Disable the grid. Enable input fields. [User Group code] & [Description]	Grid disabled. Enabled input fields.	Ok
2	Enter [!,@,#,\$,%,^,&,*] characters for [User Group Code] input field.	Disable key press for [!,@,#,\$,%,^,&,*] characters	Disabled key press for [!,@,#,\$,%,^,&,*] characters	Ok
3	Enter [A-Z, a-z & 0-9] characters for [User Group Code] input field.	Enable key press for [A-Z , a-z & 0-9] characters	Enabled key press for [A-Z , a-z & 0-9] characters	Ok
4	Enter mandatory fields value & click [Save] button	Display save status message	“Record Added Successfully” message displayed.	Ok
5	Click [Save] button without entering mandatory fields value	Display an error message	Error message displayed.	Ok
6	Save already existing [User Group Code] field value	Display an error message	“[User Group Code] Code is Already Exist” message displayed.	Ok
7	Select existing value from grid	Display value in input fields	Value displayed in input fields.	Ok
8	Select existing value from grid & click [Modify] button	Enable modifiable input fields of the selected record.	Enabled modifiable input fields of the selected record.	
9	Click [Save] button to save modified data	Display save status message	“Record updated Successfully” message displayed.	Ok
10	Select existing value from grid & click [Delete] button	Disable all input fields of the selected record & display delete confirmation message to delete the selected record. If confirmed, delete the record. Else do not delete the record.	Confirmation message displayed. If click [Yes] record delete. if click [No] record remained.	Ok
11	Click [Clear] button	Clear the input fields	input fields cleared	Ok
12	Click [Clear] button to delete a record with existing related data	Display an error message	Displayed an error message	Ok
13	Click [Exit] button	Display conformation	Confirmation message	Ok

		message box to exist from the form if confirm exist from form. Else does not exist.	displayed. If click [Yes] exist from current form. If click [No] form does not exist.	
--	--	---	---	--

Table 5.5 Test Case Descriptions – User Group

5.2 User Evaluation

Form given below (**Figure 5.1**) was answered by the users with different knowledge levels to gather the system usability and satisfactory measures of the system.

Name						
Designation						
Date	Time					
About the design and layout of our web pages:	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	
Q.1 The design (overall appearance) of the web pages is pleasing to look at.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Q.2 The choice of colours enhances the layout of the pages (e.g. the usability of the pages).	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Q.3 The layout of the pages (e.g. location of different sections) works well.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
Q.4 The images on the web pages complement the text information and are appropriate to the layout (e.g. placement, size and theme of the images).	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	

Q.5 Please provide any other comments on the website's design and layout in the text box below.

About finding your way around (navigating):	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Q.6 The overall structure of the Hotel's website is logical.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Q.7 The standard navigation bar and breadcrumb trails across the top of the pages are helpful. (The bars located directly below the University logo).	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Q.8 Using the Hotel's navigation links (e.g. the column on the left hand side of the pages) to locate major categories of information is easy and works well.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Q.9 The words used on the navigation links (e.g. the column on the left hand side of the page) are sensible and helpful.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Q.10 I easily located the information I was looking for when I came to the Hotel's web site today.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Q.11 Please provide any other comments on using the navigation in the text box below.

About the actual content of the website:	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Q.12 The content is well written and informative.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Q.13 The content is accurate.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Q.14 The content is updated and refreshed regularly.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Q.15 The words in the content are appropriate for me (e.g. commonly used language that is well understood).	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Q.16 Please provide any other comments on the content of the website in the text box below:

About using the search facility:	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Q.17 I prefer to use the search facility to find information on the site (rather than using the navigation links).	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Q.18 The search function provides relevant information.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Q.19 Please provide any other comments on using the search facility in the text box below:

Q.20 Please provide other suggestions for improving any aspect of our web site in the text box below:

We may wish to follow up points or suggestions you have made. If you are willing for us to do this please provide your name, telephone number and e-mail address. Please note that a valid email address is a mandatory part of your response.

Name:

Telephone Number :

Email:

Please select the Submit Button to send us your survey, or the Clear Button to re-enter your responses.

Figure 5.1 User Evaluation Form

The above evaluation form(**Figure 5.2**) was created and distributed across the users to collect information from the users to evaluate whether the anticipated requirements were achieved by the delivered system. This graph contains the graphical representation of the review.

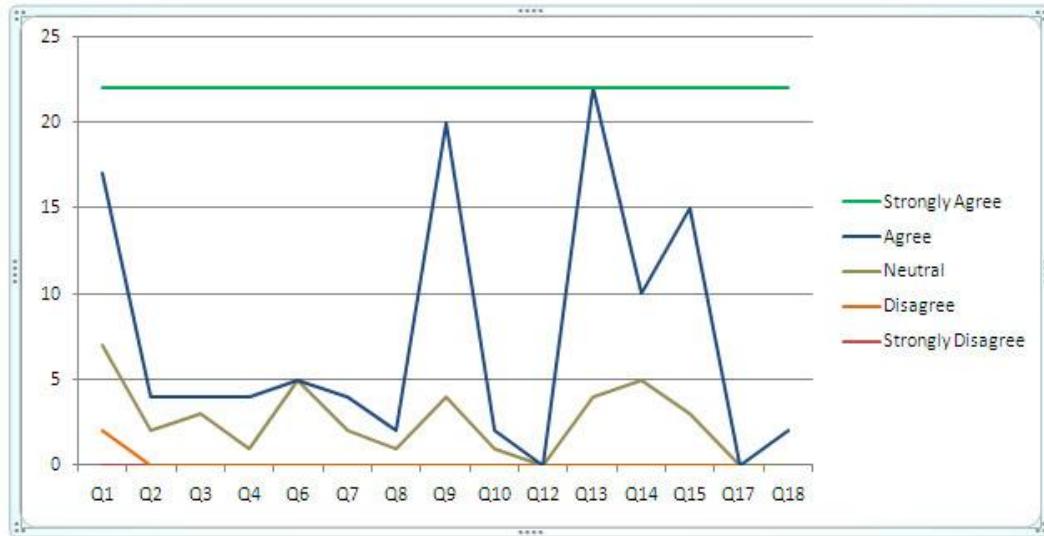


Figure 5.2 User Evaluation Form

The above (**Figure 5.2**) graph indicates the user satisfactory levels which were collected by the questionnaire.

This graph shows the future projection of system functionality depicting the expected improvements of the system.

Chapter 6 – Conclusion

6.1 Critical Assessment of Project

Almost all the objectives of the project were completed according to the user requirements, providing a basement for further enhancements as well. The requirements of the system were thoroughly analyzed with literature reviews done to improve the business and technical knowledge. The analyzed system was transformed successfully in to a design model on which the system was implemented. The analysis, design and implementation of the system were conducted conforming to object orientation. The system was tested under several stages and whatever the identified defects were rectified ensuring its performance.

The developed software enables the client organization to handle its Hotel Data, Online Room Reservation, Reservation Cost Analysis, Guest Check-in and Check-out Handling and Employee Details in an efficient and effective manner. Appropriate system security mechanisms are incorporated providing user friendly interfaces. The company would be able to attract the foreign market as well through the web site integrated with system with up-to-date and accurate details.

Although PHP, MySQL and Apache technologies were used during the semesters of BIT External Degree, the object oriented usage of PHP and MySQL were unfamiliar to me. The technical knowledge had to be gathered heavily and sample programs had to be written to get the experience. This approach has given me a lot of knowledge and experience rather than doing what I already knew.

6.2 Future enhancement of the project

- Payment Gateway Support

The payments coming from external users should be handled with the third party transaction gateway in a secured manner. Only the interface to the payment gateway has been provided by the existing system.

6.3 Leason Learnt

- The system scope should be defined clearly in the early stages of the development life cycle.
- Learnt to stick to the time schedule in the future developments
- Allocate necessary time for the system designing in the schedule

6.4 Conclusion

The project gave me a vast amount of experience and knowledge where I practiced various software engineering disciplines from the inception to the transition of the project. Almost all the project objectives were covered and the end users were provided with a usable software solution. The lessons, knowledge and the experience I gathered during this project would help me in my future career in software engineering.

References

[WWW1] PHP Tutorials www.w3schools.com [2009]

[WWW2] PHP Tutorials www.wikipedia.org [2009]

[WWW3] PHP Tutorials - www.php.net [2010]

[WWW4] Hotel Management system- www.hotsofthms.com [2009]

[WWW5] PHP Tutorials- www.phpro.org/tutorials/Object-Oriented-Programming-with-PHP.html [2009]

[WWW6] Hotel Management system- www.ezefrontdesk.com/hotel-software/features.htm [2009]

[WWW7] Email Server Download- www.ezefrontdesk.com/hotel-software/features.htm [2010]

[WWW8] Apache Tomcat Download <http://tomcat.apache.org/download-60.cgi> [2008]

[WWW9] Online Hotel Management System-
<http://www.hotelmanagementsystem.com/>

[WWW10] MySql Download <http://dev.mysql.com/downloads/mysql/5.0.html> [2008]

Appendices

Appendix A- System Documentation

The technical guidance required for HMS is presented in this appendix. Note that the explanations are focused on setting up the system on a Windows Operating System.

Setting up the Environment

The system runs on Apache-MySQL-PHP framework and this section explains how to install them in a Windows XP computer.

Installing Apache 2.0.54

- Download Apache MS Installer package (apache_2.0.54-win32-x86-no_ssl.msi) from www.apache.org
- Double click on its icon to run the installation wizard.
- Click next until reach the Server Information step.
 - Network Domain : localhost
 - Server Name : Server's full DNS name (localhost)
 - Administrator's Email Address : Server administrator's email address
- Select 'All Users, on Port 80' – the recommended option to install Apache as a Windows Service. Apache Web Server will be automatically started when the Windows XP OS is started.
- Click next button and choose Typical Installation. Specify the destination folder and complete the installation process in the next two steps.
- To check whether Apache was installed successfully, type **http://localhost** in the address bar of a web browser. A page with the Apache Logo (**Figure A.1**) would be displayed if the installation was successful.



Figure A.1 – Apache Logo

Installing PHP 5.0.5

- Download the zip file of PHP Windows installable package (**php-5.0.5-Win32.zip**) from <http://www.php.net> and extract it.

- Copy php.ini-recommended file in the extracted directory into the WINDOWS directory of the computer and rename the file as php.ini.

Apache and PHP Configuration

- Update the **httpd.conf** file in Apache configuration directory (Apache2/conf) by adding the following lines: AddType application/x-httdp-php .php
- LoadModule php5_module "<path>\php-5.0.5-Win32\php5apache2.dll"
- Restart the Apache Service.
- To check the configuration success, save a PHP file (info.php) with following code in the htdocs folder.

```
<?php phpinfo(); ?>
```

Type <http://localhost/info.php> in a web browser and a web page with the following header (**Figure A.2**) will be displayed if the configuration was successful.



Figure A.2 – PHP Info Header

Installing MySQL

- Download the zip file of MySQL Windows installable package (**mysql-5.0.xx-win.zip**) from <http://www.mysql.com> and extract it.
- Run setup.exe and keep clicking the next button to complete the installation.
- Update the php.ini file to be executed with mysqli functions.
- Restart the computer.

Installing and Configuring MySQL Administrator 5.0

- MySQL Administrator is a GUI tool that supports to do developer's operations with the MySQL database server easily. It can be downloaded from <http://dev.mysql.com/downloads/gui-tools/5.0.html>

Appendix B - Design Documentation

Table Structure

Table Name : title
 Table ID : titl
 Description : Titles

Descriptive Name of Field	Constraint/Domain Values	Field name	Data Type	Length	Not Null
Title Code	Primary Key	titl_code	char	3	Yes
Title Name		titl_name	varchar	20	Yes
Total No of Bytes				23	

Table Name : country
 Table ID : coun
 Description : Country

Descriptive Name of Field	Constraint/Domain Values	Field name	Data Type	Length	Not Null
Country Code	Primary Key	coun_code	char	3	Yes
Country Name		coun_name	varchar	20	Yes
Total No of Bytes				23	

Table Name : currency
 Table ID : curr
 Description : Currency

Descriptive Name of Field	Constraint/Domain Values	Field name	Data Type	Length	Not Null
Currency Code	Primary Key	curr_code	char	3	Yes
Currency Name		curr_name	varchar	20	Yes
Total No of Bytes				23	

Table Name : paymenttype
 Table ID : ptty
 Description : Payment Type

Descriptive Name of Field	Constraint/Domain Values	Field name	Data Type	Length	Not Null
Payment Type Code	Primary Key	ptty_code	char	3	Yes
Payment Type Name		ptty_name	varchar	20	Yes
Total No of Bytes				23	

Table Name : designation
 Table ID : desi

Description : Designation

Descriptive Name of Field	Constraint/Domain Values	Field name	Data Type	Length	Not Null
Designation Code	Primary Key	desi_code	char	3	Yes
Designation Name		desi_name	varchar	20	Yes
Total No of Bytes				23	

Table Name : department

Table ID : dept

Description : Department

Descriptive Name of Field	Constraint/Domain Values	Field name	Data Type	Length	Not Null
Department Code	Primary Key	dept_code	char	3	Yes
Department Name		dept_name	varchar	20	Yes
Total No of Bytes				23	

Table Name : floors

Table ID : flor

Description : floors

Descriptive Name of Field	Constraint/Domain Values	Field name	Data Type	Length	Not Null
Floors Code	Primary Key	flor_code	char	3	Yes
Floors Name		flor_name	varchar	20	Yes
Total No of Bytes				23	

Table Name : roomtype

Table ID : rmty

Description : Room Type

Descriptive Name of Field	Constraint/Domain Values	Field name	Data Type	Length	Not Null
Room Type Code	Primary Key	rmty_code	char	3	Yes
Room Type Name		rmty_name	varchar	20	Yes
Maximum Guest		rmty_maxgst	integer		Yes
Total No of Bytes				23	

Table Name : roominfo

Table ID : rmin

Description : Room Details

Descriptive Name of Field	Constraint/Domain Values	Field name	Data Type	Length	Not Null
Floor	Foreign Key	rmin_floor	char	3	Yes
Room Type	Foreign Key	rmin_rmtype	char	20	Yes
Room Number	Primary Key	rmin_numbr	char	3	Yes
Room Status		rmin_stat	char	1	Yes
Total No of Bytes				27	

Table Name : accommodation
 Table ID : accm
 Description : Accommodation

Descriptive Name of Field	Constraint/Domain Values	Field name	Data Type	Length	Not Null
Accommodation Code	Primary Key	accm_code	char	3	Yes
Accommodation Name		accm_name	varchar	20	Yes
Total No of Bytes				23	

Table Name : rates
 Table ID : rate
 Description : Room Details

Descriptive Name of Field	Constraint/Domain Values	Field name	Data Type	Length	Not Null
Accommodation Type	Primary Key /Foreign Key	rate_acmtype	char	3	Yes
Room Type	Primary Key /Foreign Key	rate_noofple	char	20	Yes
Amount		rate_amount	decimal		Yes
Effective Date From		rate_datefrm	date		Yes
Effective Date To		rate_dateto	date		Yes
Rate Type		rate_type	char	1	Yes
Total No of Bytes	Primary Key			23	

Table Name : mealplans
 Table ID : meal
 Description : Meal Plan

Descriptive Name of Field	Constraint/Domain Values	Field name	Data Type	Length	Not Null
Accommodation Type	Primary Key/Foreign Key	meal_acmtype	char	3	Yes

Meals		meal_desc	varchar	100	Yes
Total No of Bytes				103	

Table Name : facility
 Table ID : faci
 Description : Facilities

Descriptive Name of Field	Constraint/Domain Values	Field name	Data Type	Length	Not Null
Facility Code	Primary Key	faci_code	char	3	Yes
Facility Name		faci_name	varchar	20	Yes
Total No of Bytes				23	

Table Name : comprofile
 Table ID : comp
 Description : Company Profile

Descriptive Name of Field	Constraint/Domain Values	Field name	Data Type	Length	Not Null
Company Name		comp_name	varchar	40	Yes
Street		comp_str1	varchar	60	Yes
Street2		comp_str2	varchar	60	
City		comp_city	varchar	40	Yes
Country	Foreign Key	comp_cuntry	char	20	Yes
Primary Phone No		comp_ppno	varchar	20	Yes
Second. Phone No		comp_spno	varchar	20	
Fax		comp_fax	varchar	20	
E-mail		comp_email	varchar	40	
Company Logo		comp_logo	varchar	100	
Total No of Bytes				420	

Table Name : bookinginfo
 Table ID : bkng
 Description : Booking Infor

Descriptive Name of Field	Constraint/Domain Values	Field name	Data Type	Length	Not Null
Reservation Id	Primary Key	bkng_id	char	40	Yes
Guest Id	Primary Key	bkng_gustid	char	60	Yes
Check in Date		bkng_checkin	date		
Checkout Date		bkng_checkout	date		Yes
Room Type	Foreign Key	bkng_rmtype	char	3	Yes
Adults		bkng_adults	integer		Yes

Childs	bkng_childs	integer			
No of Rooms	bkng_noofrm	integer			
No of Days	bkng_noofday	integer			
Booking Status	bkng_status	char	1		
Reserved Date	bkng_regdate	date			
Modified Date	bkng_upddate	date			
Total No of Bytes				104	

Appendix C - User Documentation

This appendix presents necessary guidelines to operate the system. Its integrity with the company web site and other major features are explained for the users to understand the way to get the maximum use for the system.

C.1 Dream Beach Home Page

The first screen (**Figure C.1**) appears accessing the system. It serves all the necessary information regarding the system and other common details.



Figure C.1 Dream Beach Home Page

C.2 System Login Page

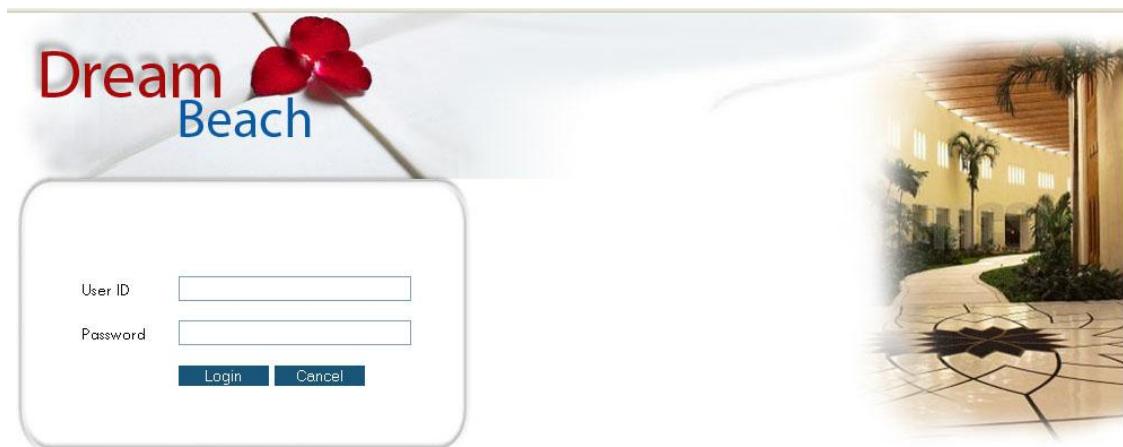


Figure C.2 System Login Page

Working with Dream beach System

Log in to the system with a valid user id and a password using the screen displayed in **Figure C.2**. Main Modules of the system (**Figure C.3**) will be displayed.

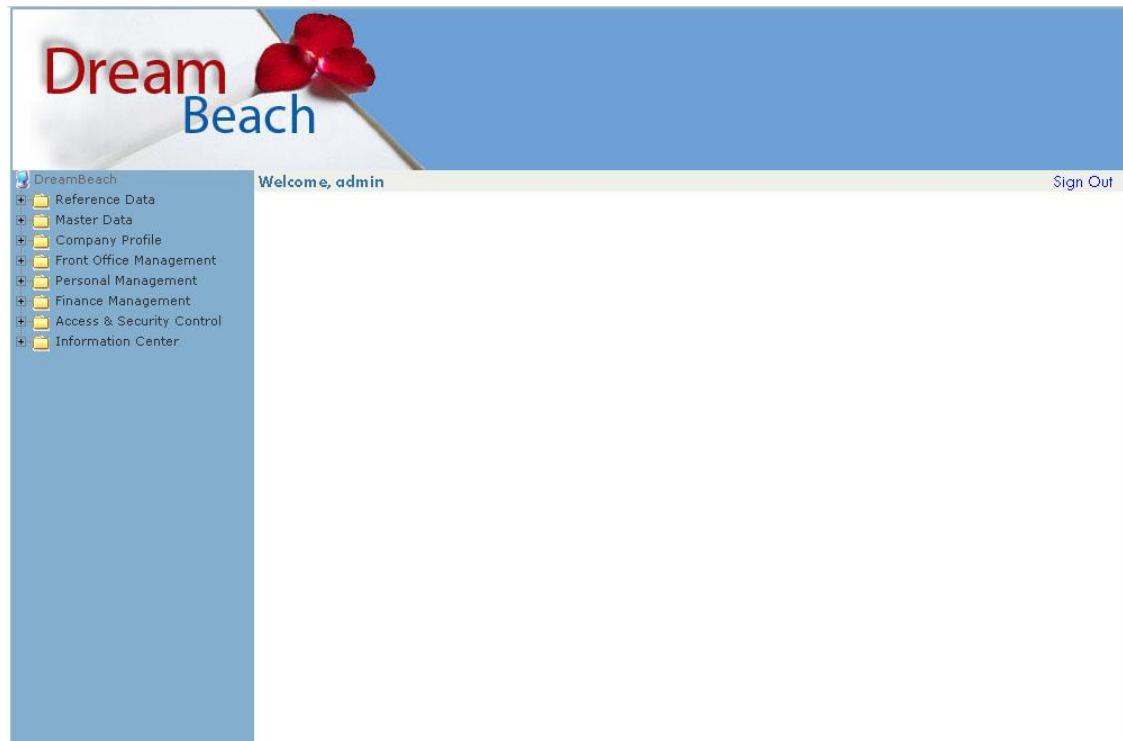


Figure C.3 Main Modules of the System

User can access each screen according to the granted user privileges. The user cannot access screens, if none of the screens in the particular module is authorized for the user to view. The module where the System Administrator can maintain these user privileges is System Security.

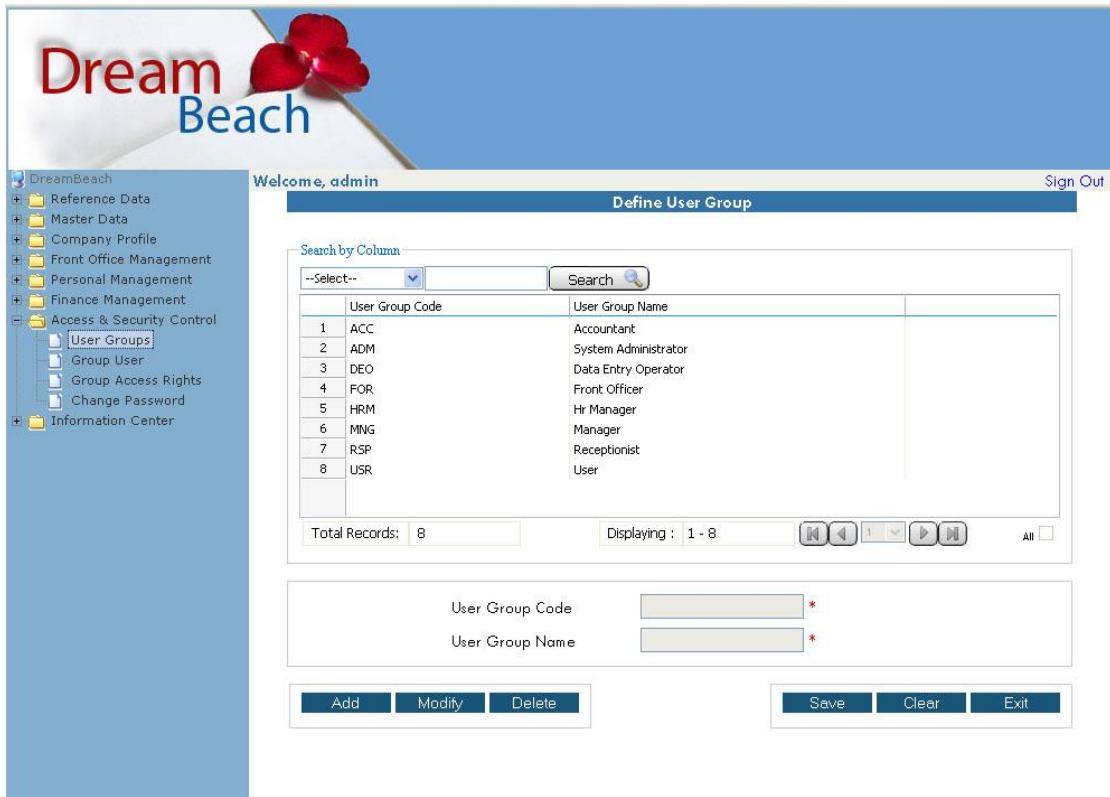


Figure C.4 Define User Group (Access /Security Module)

User Groups have to be defined in the system with a unique **User Group Code** and a **User Group Name** using the screen displayed in **(Figure C.4)**. The screen permissions can be granted for each of those groups by using ‘Group Access Rights’ screen **(Figure C.5)**.

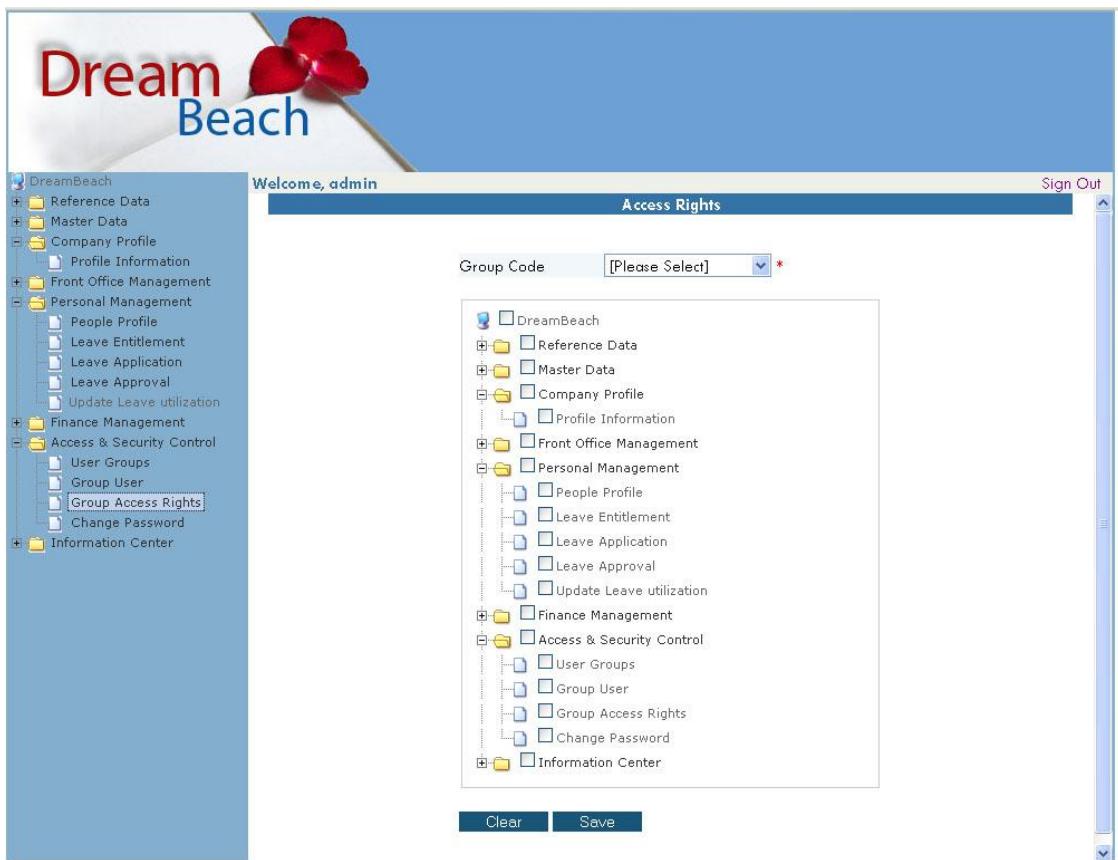


Figure C.5 Group Access Rights (Access /Security Module)

In order to create the system users, the administrator has to define the employees of the company by using the screen provided in **Personal Management** module. (**Figure C.6**)

The screenshot shows the Dream Beach Hotel software interface. The top bar displays the logo 'Dream Beach'. The left sidebar contains a navigation menu with the following items:

- Reference Data
- Master Data
- Company Profile
- Front Office Management
- Personal Management (selected)
- Finance Management
- Access & Security Control
- User Groups
- Group User
- Group Access Rights
- Change Password
- Information Center

The main content area is titled 'Welcome, admin'. It shows a 'Personel Info.' form with the following fields:

Employee No	*
Title	[Please Select] *
Name With Initials	*
Full Name	*
First Name	*
Last Name	*
User Name	admin Check Availability
Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female
Date of Birth	*
Age	
Marital Status	<input checked="" type="radio"/> Single <input type="radio"/> Married
NIC No	*
People Category	[Please Select] *
Department	[Please Select] *
Designation	[Please Select] *
Date of Join	*

At the bottom of the form are buttons for 'Add', 'Modify', 'Inactive', 'Save', 'Clear', and 'Exit'. The footer of the page reads '© 2009 Dream Beach Hotel (Pvt) Ltd. All Rights Reserved'.

Figure C.6 People Profile (Personal Management Module)

Each Employee has to be defined with a unique **Employee Id**, **User Name** and temporary **password**. Then these employees can be mapped to the defined user groups with different user privileges by using the **Group Users** screen (**Figure C.7**).

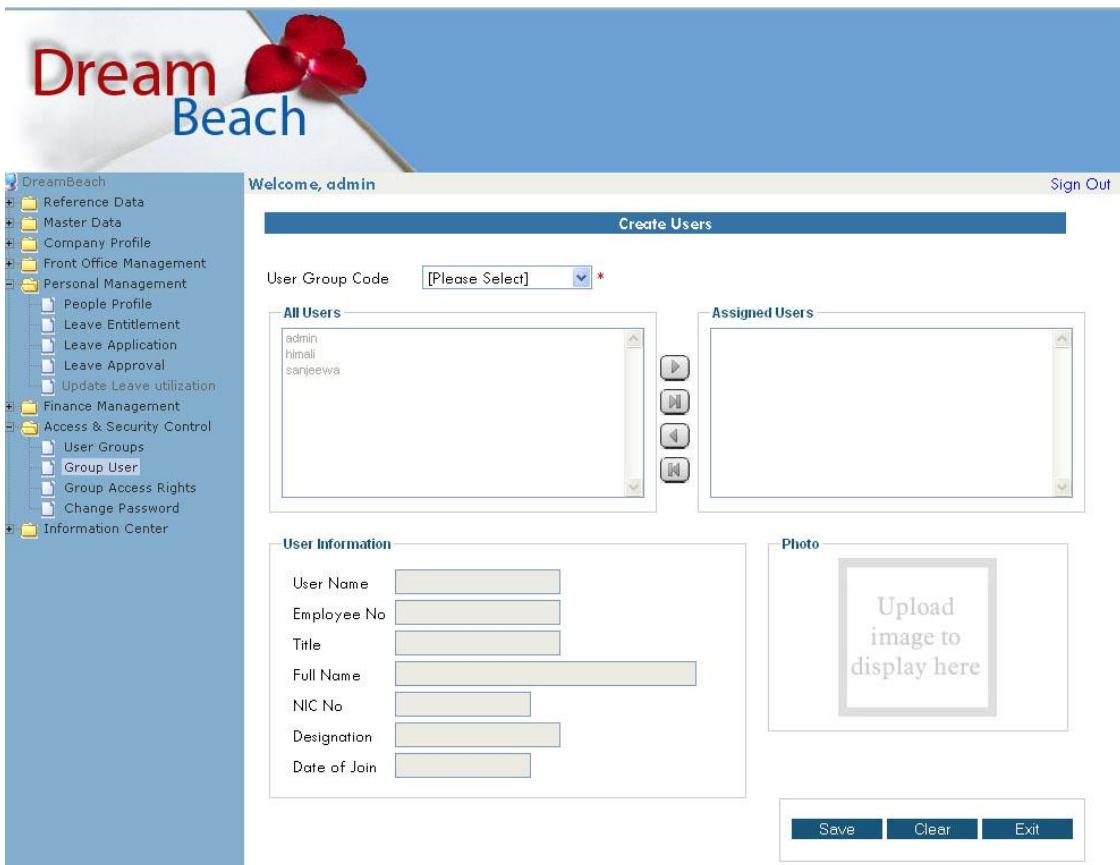


Figure C.7 Group User (Access/Security Module)

The created users can log in to the system using temporary password and change the password through the provided screen (**Figure C.8**).

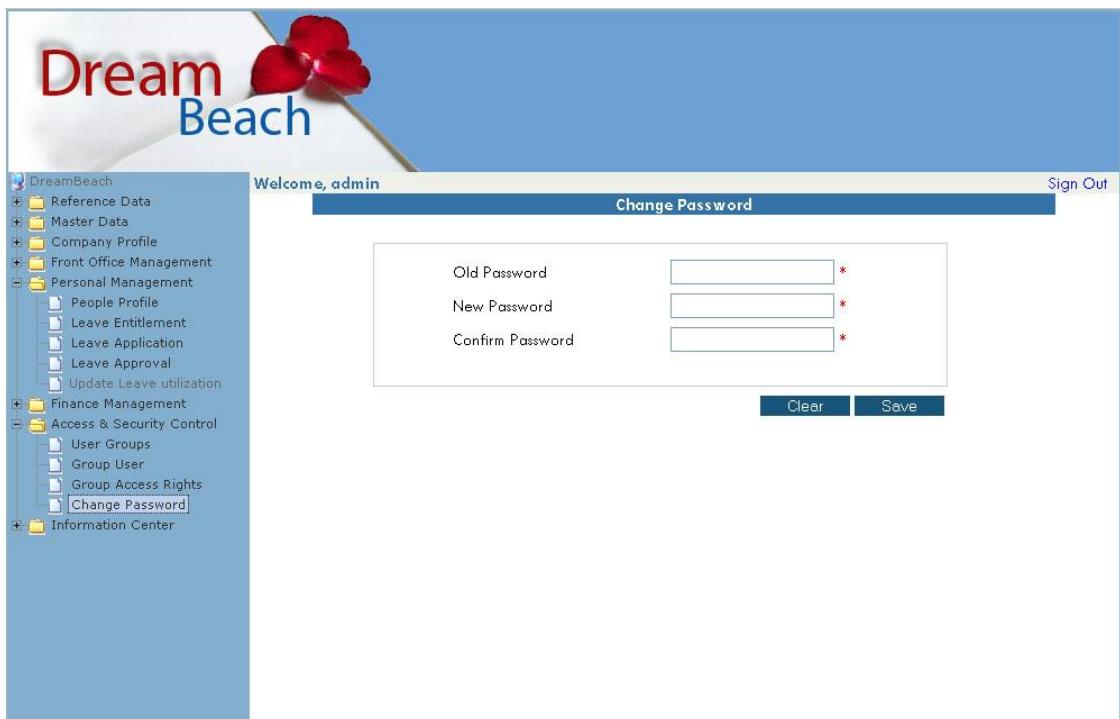


Figure C.8 Change Password (Access/Security Module)

Company profile can be updated using the screen provided in Company Profile Module (**Figure C.9**). All the basic information about the company can be maintained using this screen.

The screenshot shows the 'Company Profile' screen for 'Dream Beach'. The left sidebar lists modules: Reference Data, Master Data, Company Profile (selected), Profile Information, Front Office Management, Personal Management, Finance Management, Access & Security Control, and Information Center. The main area displays company details: Company Name (Dream Beach Resort), Street (No:506,Galle Road, Ambalangoda), City (Ambalangoda), Country (Sri Lanka), Primary Phone No (0912258873), Secondary Phone No (empty), Fax (empty), and E-mail (dbra@slt.lk). A logo for 'DREAM BEACH RESORT SRI LANKA' is displayed, along with a 'Browse...' button for file uploads. At the bottom are 'Save', 'Clear', and 'Exit' buttons.

Figure C.9 Profile Information (Company Profile)

Reference data module defines all the reference data which are needed in the system references.

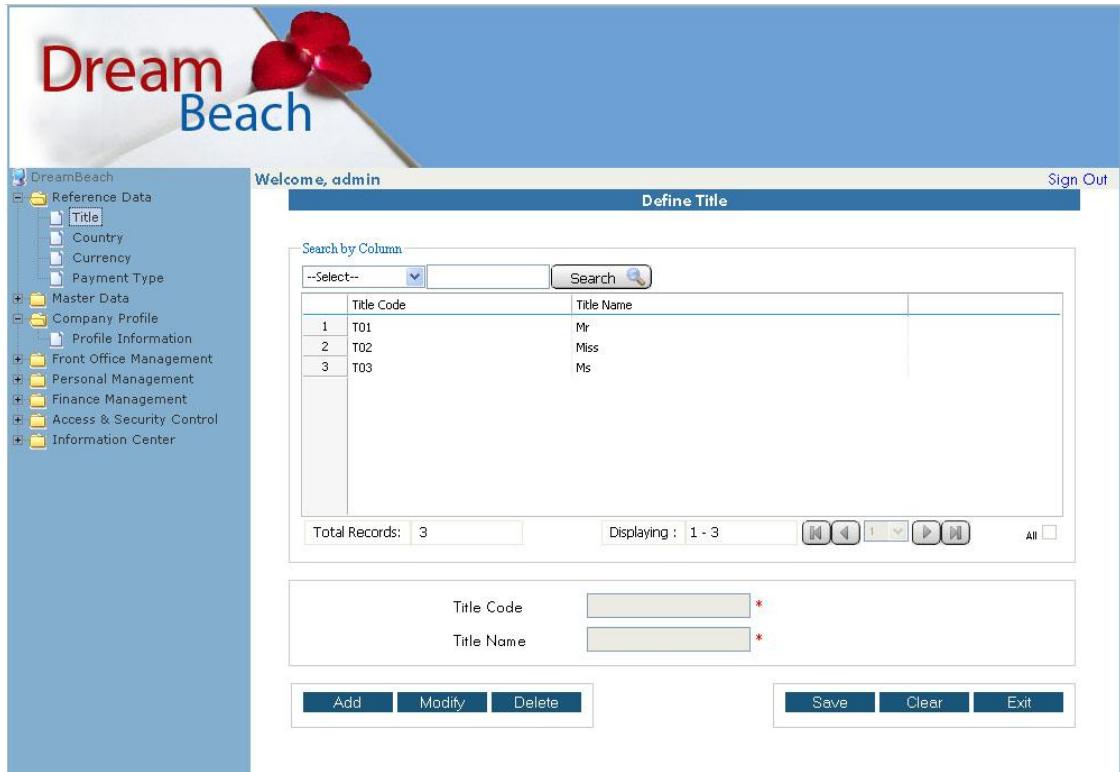


Figure C.10 Define Title (Reference Data Module)

This user interface (**Figure C.10**) provides the facility to system user to define Titles using Title Code and Title Name. Also, the details of existing Titles can be modified and deleted using this interface.

Master data module defines all the reference data which are needed in the system transactions.

The screenshot shows the Dream Beach software interface. The title bar reads "Dream Beach". The left sidebar contains a navigation menu with categories like Reference Data, Master Data, Company Profile, Front Office Management, Personal Management, Finance Management, Access & Security Control, and Information Center. Under "Master Data", "Room Type" is selected, which is further expanded to show "Room Information". The main content area is titled "Define Room" and displays a table of room information. The table has columns: Floor, Room Type, Room No, and Room Status. The data in the table is as follows:

	Floor	Room Type	Room No	Room Status
1	First Floor	Double	102	A
2	First Floor	Double	103	A
3	Second Floor	Double	202	A
4	Second Floor	Double	203	A
5	Third Floor	Double	302	A
6	Third Floor	Double	303	A
7	Fourth Floor	Double	402	A
8	Fourth Floor	Double	403	A
9	First Floor	Single	101	A
10	Second Floor	Single	201	A

Below the table, there are search and pagination controls: "Total Records: 20", "Displaying: 1 - 10", and navigation buttons. The form below the table includes fields for "Floor" (dropdown, required), "Room Type" (dropdown, required), "Room No" (text input, required), and status checkboxes for "Activate" and "Deactivate". At the bottom are buttons for "Add", "Modify", "Delete", "Save", "Clear", and "Exit".

Figure C.11 Room Information (Master Data Module)

This user interface (**Figure C.11**) provides the facility to system user to define room using Floor Type, Room Type Name, Room Number and status of the room. Also, the status of existing Room can be modified using this interface.

Appendix D – Managements Reports

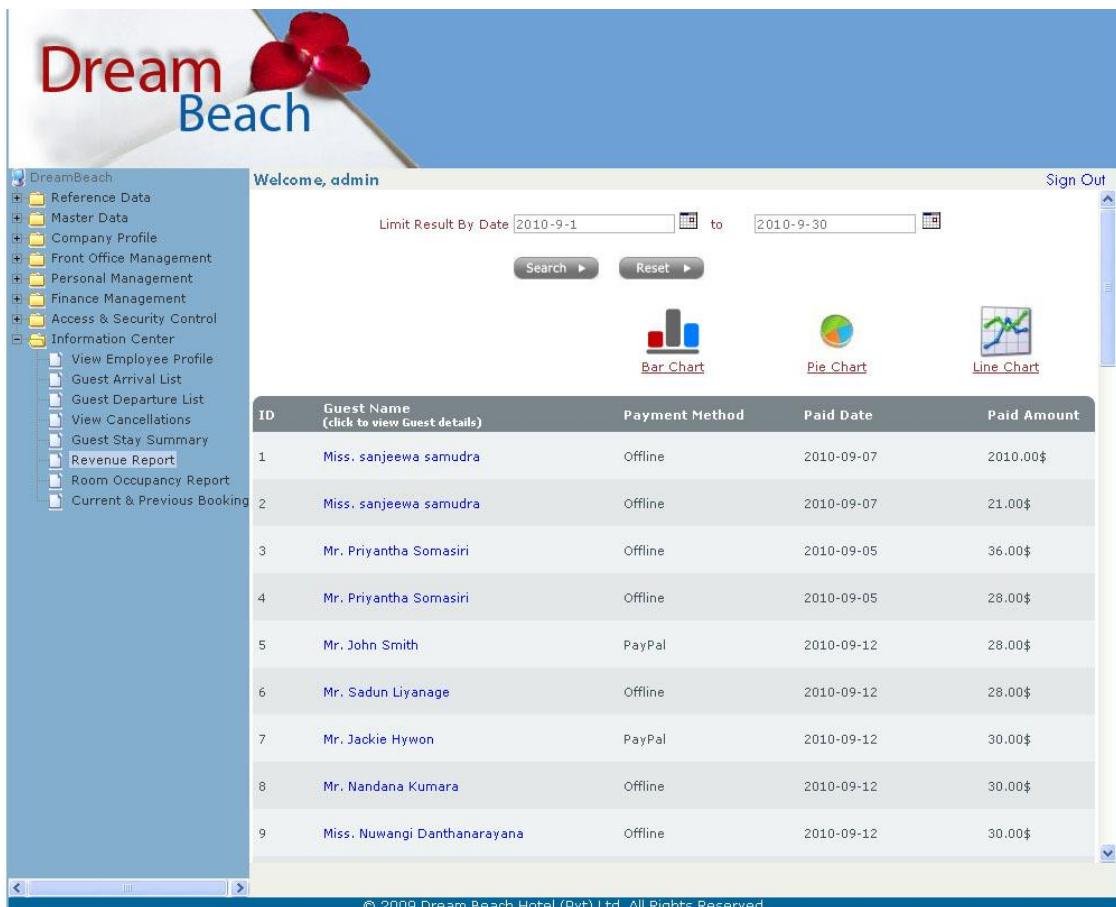


Figure D.1- Revenue Reports (Information Center)

This user interface (**Figure D.1**) provides a report which containing the income of the company within a user selected period of time. The graphical views of the report also provided as a bar chart and Line Chart (**Figure D.2 and Figure D.3**).

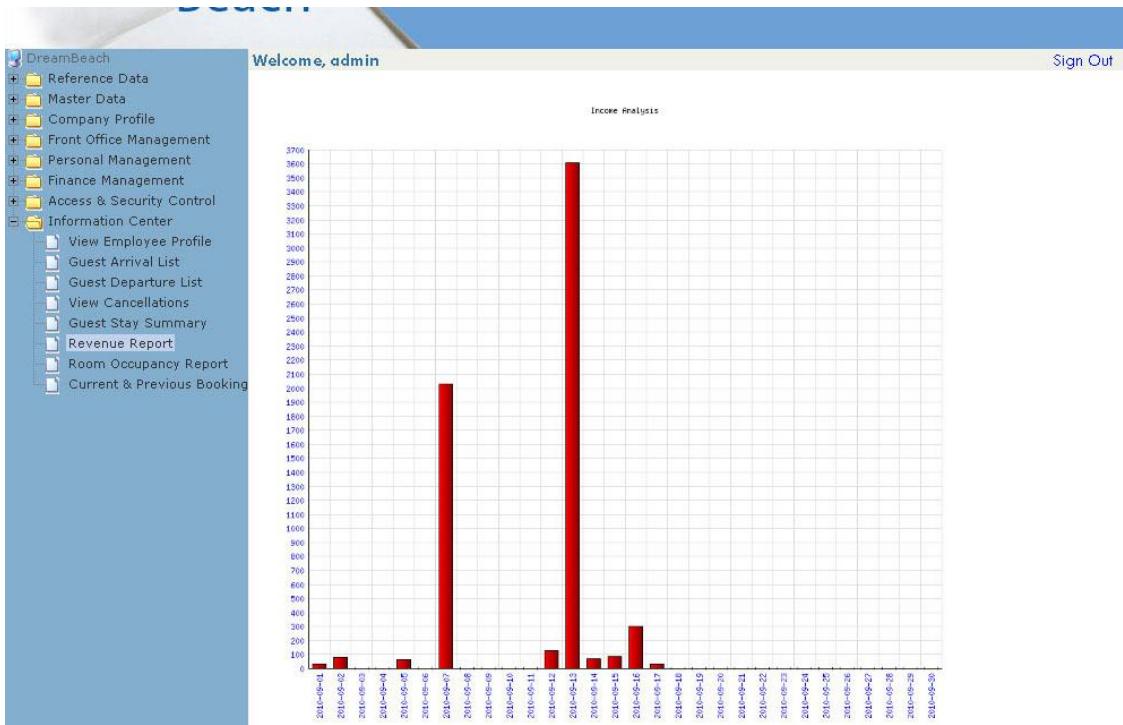


Figure D.2 Revenue Report as a Bar Chart

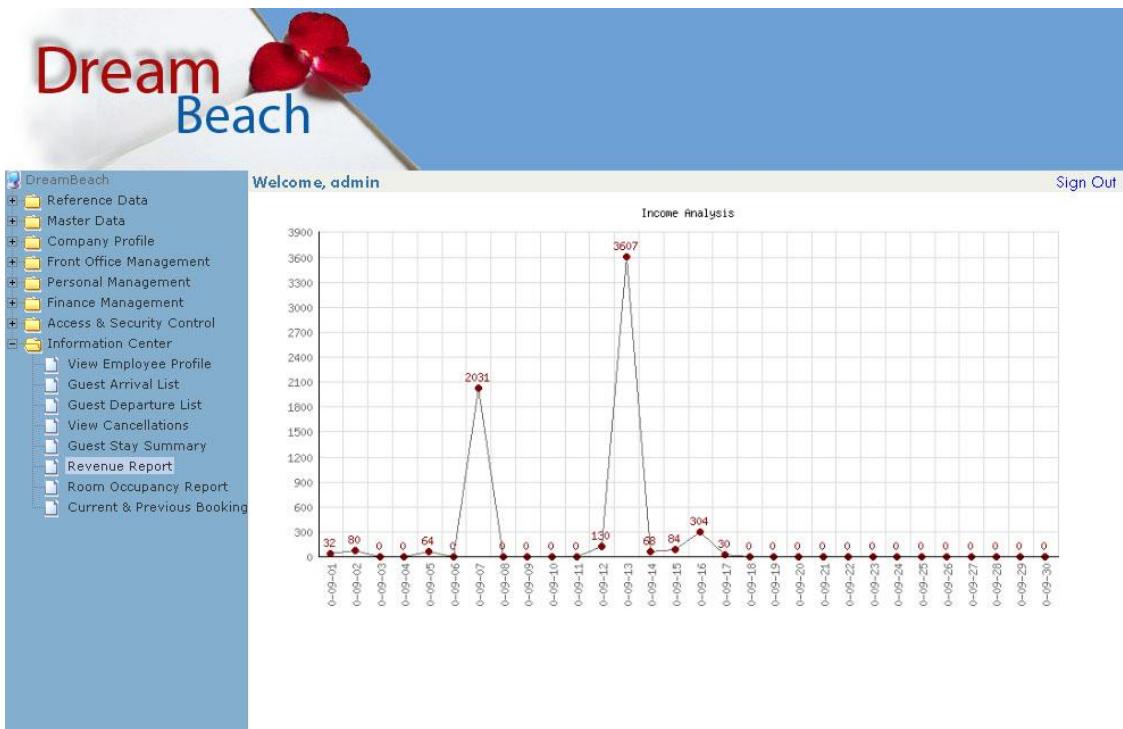


Figure D.3 Revenue Report as a Line Chart

The screenshot shows a web-based application for managing hotel bookings. The header features the "Dream Beach" logo with a red rose graphic. The left sidebar contains a navigation menu with categories like Reference Data, Master Data, Company Profile, Front Office Management, Personal Management, Finance Management, Access & Security Control, and Information Center. Under the Information Center, there are links for View Employee Profile, Guest Arrival List, Guest Departure List, View Cancellations, Guest Stay Summary, Revenue Report, Room Occupancy Report, and Current & Previous Booking. The main content area is titled "Welcome, admin". It includes search filters for "Limit Result By Guest" (text input), "Limit Result By Country" (dropdown set to "Please Select"), "Limit Result By Date" (date range inputs), and buttons for "Submit" and "Reset". Below these is a table with columns: ID, Guest Name (click to view Guest details), Arrival Date, Departure Date, and Room Type. The table lists guest arrivals from September 18, 2010, to September 22, 2010, with room types ranging from Single to Triple. At the bottom of the page is a copyright notice: "© 2009 Dream Beach Hotel (Pvt) Ltd. All Rights Reserved".

ID	Guest Name (click to view Guest details)	Arrival Date	Departure Date	Room Type
4	Mr. Priyantha Somasiri	2010-09-27	2010-09-30	Single
15	Mr. Robinson Cruso	2010-09-28	2010-09-30	Double
25	Mr. Chan Jackie	2010-09-18	2010-09-22	Double
26	Mr. Jony Marcony	2010-09-18	2010-09-25	Single
39	Mr. akila kodithuwakku	2010-09-10	2010-09-11	Single
51	Mr. Tharanga Kumara	2010-09-18	2010-09-19	Triple
53	Mr. Priyankara Perera	2010-09-18	2010-09-20	Triple
56	Mr. Janaka Rabukwella	2010-09-18	2010-09-29	Single
57	Mr. Jenifer Anthoney	2010-09-18	2010-09-19	Triple
58	Mr. Tony Eric	2010-09-18	2010-09-22	Triple

Figure D.4 Guest Arrival List

This user interface (**Figure D.4**) provides a report which containing the Guest Arrival List within a selected period of time.

This screenshot shows the same guest arrival list interface as Figure D.4, but with a different set of guest arrivals listed. The table below shows the new data.

ID	Guest Name (click to view Guest details)	Arrival Date	Departure Date	Room Type
1	Miss. sanjeewa samudra	2010-09-01	2010-09-04	Single
36	Mr. Mark Anthoney	2010-09-06	2010-09-10	Triple
37	Mr. Vijith Suranga	2010-09-08	2010-09-10	Single
38	Mr. Anurudha LankaNanda	2010-09-01	2010-09-02	Single
40	Mr. Vishva Kumara	2010-08-23	2010-08-26	Double
41	Ms. Kanchana Disanayaka	2010-08-24	2010-08-25	Double
42	Ms. Kanchanamala Thennakon	2010-08-24	2010-08-25	Double
43	Mr. Roshan Dikwella	2010-08-24	2010-08-26	Single
44	Mr. Ruwan Rabukwella	2010-08-03	2010-08-06	Triple
45	Ms. Anjali Padukon	2010-08-10	2010-08-13	Single

Figure D.5 Guest Departure List

This user interface (**Figure D.5**) provides a report which containing the Guest Departure List within a user selected period of time.

The screenshot shows a web-based application for managing guest departures at Dream Beach. The header features the 'Dream Beach' logo with three red cherries. The left sidebar contains a navigation menu with categories like Reference Data, Master Data, Company Profile, etc., and a specific section for 'Information Center' which includes 'View Cancellations'. The main content area is titled 'Welcome, admin' and displays a search form with fields for 'Limit Result By Guest' (text input), 'Limit Result By Country' (dropdown set to 'Please Select'), 'Limit Result By Date' (date range inputs), and buttons for 'Search' and 'Reset'. Below the search form is a table listing guest departure details:

ID	Guest Name (click to view Guest details)	Arrival Date	Departure Date	Room Type
5	Mr. John Smith	2010-08-02	2010-08-06	Double
6	Mr. Sadun Liyanage	2010-09-20	2010-09-22	Single
8	Mr. Nandana Kumara	2010-10-11	2010-10-13	Triple
9	Miss. Nuwangi Danthanarayana	2010-10-18	2010-10-21	Single
10	Mr. Chaminda Bandara	2010-10-25	2010-10-28	Double
11	Miss. Romeka hunsun	2010-10-01	2010-10-03	Single
12	Mr. Romesh khan	2010-11-02	2010-11-06	Triple
14	Mr. Minyork Huyan	2010-10-11	2010-10-13	Single
16	Miss. Kathie Siera	2010-09-27	2010-09-30	Single
17	Mr. Dilshan Kanna	2010-12-06	2010-12-09	Single

At the bottom of the page, there is a copyright notice: © 2009 Dream Beach Hotel (Pvt) Ltd. All Rights Reserved.

Figure D.6 Reservation Cancellation

This user interface (**Figure D.6**) provides a report which containing the Reservation cancellation details within the user selected period of time.

Appendix E – Test Results

Test cases given below (**Table E.1**) will look after the scenarios in the Define Country function of the Reference Data Module.

Test Case Id		6		
Tested Component		Define Country		
No	Test Case	Test Case Description		
		Expected Output	Actual Output	Test Status
1	Click [Add] button	Disable the grid. Enable input fields. [Country Code] and [Country Name]	Grid disabled. Enabled input fields.	Ok
2	Enter [!,@,#,\$,%,&,*] characters for [Country Code] input field.	Disable key press for [!,@,#,\$,%,&,*] characters	Disabled key press for [!,@,#,\$,%,&,*] characters	Ok
3	Enter [A-Z, a-z & 0-9] characters for [Country Code] input field.	Enable key press for [A-Z , a-z & 0-9] characters	Enabled key press for [A-Z , a-z & 0-9] characters	Ok
4	Enter mandatory fields value & click [Save] button	Display save status message	“Record Added Successfully” message displayed.	Ok
5	Click [Save] button without entering mandatory fields value	Display an error message	Error message displayed.	Ok
6	Save already existing [Country Code] field value	Display an error message	“[Country Code] Code is Already Exist” message displayed.	Ok
7	Select existing value from grid	Display value in input fields	Value displayed in input fields.	Ok
8	Select existing value from grid & click [Modify] button	Enable modifiable input fields of the selected record.	Enabled modifiable input fields of the selected record.	
9	Click [Save] button to save modified data	Display save status message	“Record updated Successfully” message displayed.	Ok
10	Select existing value from grid & click [Delete] button	Disable all input fields of the selected record & display delete confirmation message to delete the selected record. If confirmed, delete the record. Else do not delete the record.	Confirmation message displayed. If click [Yes] record delete. if click [No] record remained.	Ok
11	Click [Clear] button	Clear the input fields	input fields cleared	
12	Click [Exit] button	Display conformation message box to exist from the form. if confirm exist from form. Else does not exist.	Confirmation message displayed. If click [Yes] exist from current form. If click [No] form does not exist.	

Table E.1 Test Case Descriptions – Define Country

Test cases given below (**Table E.2**) will look after the scenarios in the Accommodation Type function of the Master Data Module.

Test Case Id		7		
Tested Component		Accommodation Type		
No	Test Case	Test Case Description		
		Expected Output	Actual Output	Test Status
1	Click [Add] button	Disable the grid. Enable input fields. [Accommodation Type Code] and [Accommodation Type Name]	Grid disabled. Enabled input fields.	Ok
2	Enter [!,@,#,\$,%,&,*] characters for [Accommodation Type Code] input field.	Disable key press for [!,@,#,\$,%,&,*] characters	Disabled key press for [!,@,#,\$,%,&,*] characters	Ok
3	Enter [A-Z, a-z & 0-9] characters for [Accommodation Type Code] input field.	Enable key press for [A-Z , a-z & 0-9] characters	Enabled key press for [A-Z , a-z & 0-9] characters	Ok
4	Enter mandatory fields value & click [Save] button	Display save status message	“Record Added Successfully” message displayed.	Ok
5	Click [Save] button without entering mandatory fields value	Display an error message	Error message displayed.	Ok
6	Save already existing [Accommodation Type Code] field value	Display an error message	“[Accommodation Type Code] Code is Already Exist” message displayed.	Ok
7	Select existing value from grid	Display value in input fields	Value displayed in input fields.	Ok
8	Select existing value from grid & click [Modify] button	Enable modifiable input fields of the selected record.	Enabled modifiable input fields of the selected record.	
9	Click [Save] button to save modified data	Display save status message	“Record updated Successfully” message displayed.	Ok
10	Select existing value from grid & click [Delete] button	Disable all input fields of the selected record & display delete confirmation message to delete the selected record. If confirmed, delete the record. Else do not delete the record.	Confirmation message displayed. If click [Yes] record delete. if click [No] record remained.	Ok
11	Click [Clear] button	Clear the input fields	input fields cleared	

12	Click [Exit] button	Display conformation message box to exist from the form. If confirm exist from form. Else does not exist.	Confirmation message displayed. If click [Yes] exist from current form. If click [No] form does not exist.	
----	---------------------	---	--	--

Table E.2 Test Case Descriptions – Accommodation Type

Test cases given below (**Table E.3**) will look after the scenarios in the Floor Type function of the Master Data Module.

Test Case Id		8		
Tested Component		Define Floor		
Test Case Description				
No	Test Case	Expected Output	Actual Output	Test Status
1	Click [Add] button	Disable the grid. Enable input fields. [Floor Code] and [Floor Name]	Grid disabled. Enabled input fields.	Ok
2	Enter [!,@,#,\$,%,^,&,*] characters for [Floor Code] input field.	Disable key press for [!,@,#,\$,%,^,&,*] characters	Disabled key press for [!,@,#,\$,%,^,&,*] characters	Ok
3	Enter [A-Z, a-z & 0-9] characters for [Floor Code] input field.	Enable key press for [A-Z , a-z & 0-9] characters	Enabled key press for [A-Z , a-z & 0-9] characters	Ok
4	Enter mandatory fields value & click [Save] button	Display save status message	“Record Added Successfully” message displayed.	Ok
5	Click [Save] button without entering mandatory fields value	Display an error message	Error message displayed.	Ok
6	Save already existing [Floor Code] field value	Display an error message	“[Floor Code] Code is Already Exist” message displayed.	Ok
7	Select existing value from grid	Display value in input fields	Value displayed in input fields.	Ok
8	Select existing value from grid & click [Modify] button	Enable modifiable input fields of the selected record.	Enabled modifiable input fields of the selected record.	
9	Click [Save] button to save modified data	Display save status message	“Record updated Successfully” message displayed.	Ok
10	Select existing value from grid & click [Delete] button	Disable all input fields of the selected record & display delete confirmation message to delete the selected record. If confirmed, delete the record. Else do not delete the record.	Confirmation message displayed. If click [Yes] record delete. if click [No] record remained.	Ok
11	Click [Clear] button	Clear the input fields	input fields cleared	

12	Click [Exit] button	Display conformation message box to exist from the form. If confirm exist from form. Else does not exist.	Confirmation message displayed. If click [Yes] exist from current form. If click [No] form does not exist.	
----	---------------------	---	--	--

Table E.3 Test Case Descriptions – Define Floor

Following test cases (**Table E.4**) will look after the scenarios in the Company Profile function of the Company Profile Module.

Test Case Id		9		
Tested Component		Company Profile		
No	Test Case	Expected Output	Actual Output	Test Status
1	Enter [A-Z,!,@,#,\$,%,^,&,*] characters for [Company Name, Street, City and Email] input fields.	Enabled key press for [A-Z,!,@,#,\$,%,^,&,*] characters	Enabled key press for [!,@,#,\$,%,^,&,*] characters	Ok
2	Enter [+ & 0-9] characters for [Primary Phone Number, Secondary Phone Number and Fax] input fields.	Disabled key press for [A-Z , a-z & ,!,@,#,\$,%,^,&,*] characters	Disabled key press for [A-Z , a-z & ,!,@,#,\$,%,^,&,*] characters	Ok
3	Enter mandatory fields value & click [Save] button	Display save status message	“Record Added Successfully” message displayed.	Ok
5	Click [Save] button without entering mandatory fields value	Display an error message	Error message displayed.	Ok
	Click [Save] button without entering valid Email Address	Display an error message	Error message displayed.	Ok
9	Click [Save] button to save modified data	Display save status message	“Record updated Successfully” message displayed.	Ok
11	Click [Clear] button	Clear the input fields	input fields cleared	
12	Click [Exit] button	Display conformation message box to exist from the form. If confirm exist from form. Else does not exist.	Confirmation message displayed. If click [Yes] exist from current form. If click [No] form does not exist.	

Table E.4 Test Case Descriptions – Company Profile

Appendix F - Code Listing

This appendix will be summarized several significant code segments which have been integrated in to proposed system.

Reference Data Module

Define Title

add_title.php : client side and web tier coding with HTML ,Java Script and PHP

```
<?php
ob_start();
session_start();
$v = $_SESSION['path'] = pathinfo($_SERVER['PHP_SELF']);
include("../includes/config.php");
include("../includes/errors.php");
include("../dbconn.php");
include("security.php");
$con=new ConnectionDB();
include("title.php");

$programme_name="Define Title";
$label1="Title Code";
$label2="Title Name";

if($_SERVER['REQUEST_METHOD'] == "POST"){

    $titleCode = trim($_REQUEST["titleCode"]);
    $titleName = trim($_REQUEST["titleName"]);
    $hidProcess = trim($_REQUEST["hidProcess"]);
    $err_no =0;

    //empty fields validation

    if($titleCode==""){
        $err_no++;
        $error_msg1=validateMandatory($label1);
    }
    if($titleName==""){
        $err_no++;
        $error_msg2=validateMandatory($label2);
    }
    if($err_no==0){
        //assign request fields to an array
        $arrayVal=array($titleCode,$titleName);
        $tit=new Title($arrayVal);
    }
    //call function to save records
    if($hidProcess=="add"){
        $msg=$tit->save();
```

```

        }
//call function to modify data
        if($hidProcess=="alter"){
            $msg=$tit->update();
        }
//call function to delete data
        if($hidProcess=="remove"){
            $msg=$tit->remove();
        }
    }
?>

<?php
    include("../includes/grid.php");
?>
<script src="../js/validation.js"></script>

<script language="javascript" type="text/javascript">

//function to enable text fields
    function enableFileds(){
        document.getElementById("titleCode").disabled = false;
        document.getElementById("titleName").disabled = false;
    }

//function to disable text fields
    function disableFileds(){
        document.getElementById("titleCode").disabled = true;
        document.getElementById("titleName").disabled = true;
    }

//function to Add button
    function btnAdd_Click(){
        clearAll();
        document.getElementById("btnAdd").disabled=true;
        document.getElementById("btnModify").disabled=true;
        document.getElementById("btnDelete").disabled=true;
        document.getElementById("btnSave").disabled=false;
        document.getElementById("btnClear").disabled=false;
        enableFileds();
        myObj1.disabled();
        document.getElementById("hidProcess").value="add";
    }

//function to clear text fields
    function btnClear_Click(){
        clearAll();
        document.getElementById("btnAdd").disabled=false;
        document.getElementById("btnSave").disabled=true;
        document.getElementById("btnClear").disabled=true;
        document.getElementById("titleCode").disabled = true;
        document.getElementById("titleName").disabled = true;
        myObj1.enabled();
    }

```

```

        }

function clearAll(){
    document.getElementById("titleCode").value="";
    document.getElementById("titleName").value="";
}

//function to modify button
function btnModify_Click(){
    document.getElementById("titleName").disabled=false;
    document.getElementById("btnSave").disabled=false;
    document.getElementById("btnClear").disabled=false;
    document.getElementById("btnAdd").disabled=true;
    document.getElementById("btnModify").disabled=true;
    document.getElementById("btnDelete").disabled=true;
    document.getElementById("hidProcess").value="alter";
}

//Function to Exit & Delete Confirm
function Confirm(para){
    if(para=="Exit"){
        var conf=confirm("Do you want to Exit?");
        if(conf==true){
            document.location='blank.php'
            return true;
        }
        else{
            return false;
        }
    }
    if(para=="Delete"){
        var conf=confirm("Do you want to delete this record?");
        if(conf==true){
            enabledFields();
            document.getElementById("hidProcess").value="remove";
            document.frmTitle.submit();
            return true;
        }
        else{
            return false;
        }
    }
}

```

</script>

```

<form name="frmTitle" method="post" action=<?php echo $PHP_SELF; ?>?<?php echo
$para?>" onsubmit="enableFileds()">
    <td width=75% valign="top" align="center" style="padding-left:100px">
        <table cellpadding="0" cellspacing="0" border="0" width="96%" align="center">
            <tr>
                <td valign="top">
                    <table cellpadding="0" cellspacing="0" align="center" border="0" width="95%">
                        <tr height="5px"><td class="prgHeading" bgcolor="#3573a6"><?php echo
$programme_name?></td></tr>
                        <tr height="5px"><td></td></tr>

```

```

<tr height="10px"><td colspan="2" class="msg_td"><?php echo $msg;?></td></tr>
    <tr height="10px"><td colspan="2"></td></tr>
</table>
</td>
</tr>
<tr>
    <td valign="top">
        <table cellpadding="0" cellspacing="0" align="center" border="0" width="100%" bgcolor="#FFFFFF">
            <tr>
                <td align="center">
                    <style>
                        #firstGrid .aw-column-0 { width: 200px; }
                        #firstGrid .aw-column-1 { width: 300px; }
                        #firstGrid .aw-grid-cell { border-right: 1px solid threedlightshadow; }
                        #firstGrid .aw-grid-row { border-bottom: 1px solid threedlightshadow; }
                        /* box model fix for strict doctypes, safari */
                        .aw-strict #myGrid .aw-grid-cell { padding-right: 3px; }
                        .aw-strict #myGrid .aw-grid-row { padding-bottom: 3px; }
                    </style>
<script>
<?php
    $query_allrecords      = "select titl_code,titl_name from title";
    $result_allrecords     = mysql_query($query_allrecords);
    $rows_allrecords        = mysql_num_rows($result_allrecords);
    $query_select          = "select titl_code,titl_name from title limit 10 offset 0";
    $result_select          = mysql_query($query_select);
    $rows_select            = mysql_num_rows($result_select);
    $rows_select            = $rows_allrecords;
    $record=""';
    while($record_select = mysql_fetch_array($result_select)){
        $record.= "[{$record_select[0]},{$record_select[1]}],";
    }
    $record.= "[]";
?
var myColumns           = ["<?php echo $label1?>", "<?php echo $label2?>,"];
var toolbarURL;
var gridLegend = 'Search by Column';
var gridSelect           = '--Select--';
var gridSelectMsg        ='Please select a search field.' ;
var searchBtn            = 'Search';
var lbltotalRecords = 'Total Records:';
var display              = 'Displaying :';
var gridAll               = 'All';
var param                = '<?php echo $query_allrecords;?>';
var myData               = [<?php echo $record?>];
var str                 = new AW.Formats.String;
var cellFormat           = [str,str];
var Records              = <?php echo $rows_select?>;
var myObj1               = new AW.UI.Grid;
createGrid(myObj1
,'myObj1','firstGrid',myData,myColumns,1,cellFormat,Records,10,'handleGridAjaxRequests.php',param);

```

```

myObj1.onSelectedRowsChanged = function(row){
    document.getElementById("titleCode").value=this.getText(0,row);
    document.getElementById("titleName").value=this.getText(1,row);
    document.getElementById("titleCode").disabled=true;
    document.getElementById("titleName").disabled=true;
    document.getElementById("btnModify").disabled=false;
    document.getElementById("btnDelete").disabled=false;
}
</script>
</td>
</tr>
</table>
</td>
</tr>
<tr height="15px"><td colspan="2"></td></tr>
<tr>
    <td valign="top">
        <table cellpadding="0" cellspacing="0" align="center" border="0" width="100%">
            <tr>
                <td style="" valign="top" align="center">
                    <fieldset style="width:680">
                        <table cellpadding="0" cellspacing="0" align="center" width="60%" class="inputdata_tbl">
                            <tr height="6px"><td colspan="2"></td></tr>
                            <tr>
                                <td><?php echo $label1?></td>
                                <td><input type="text" id="titleCode" name="titleCode" maxlength="3" onkeyup="toUpperCase(this.id)" onchange="toUpperCase(this.id)" onkeypress="return CharacNemariconly(event)" disabled="disabled" onBlur="validate('titleCode','msgCode','<?php echo validateMandatory($label1)?>')" value="<?php echo $titleCode?>"><span class="req">&ampnbsp*</span></td>
                            </tr>
                            <tr>
                                <td colspan="2" id="msgCode" style="color:#FF0000; font-family:Arial, Helvetica, sans-serif; font-size:10px;text-align:center"><?php echo $error_msg1;?></td>
                            </tr>
                            <tr height="8px"><td colspan="2"></td></tr>
                            <tr>
                                <td><?php echo $label2?></td>
                                <td><input type="text" id="titleName" name="titleName" maxlength="40" disabled="disabled" onBlur="validate('titleName','msgName','<?php echo validateMandatory($label2)?>')" value="<?php echo $titleName?>"><span class="req">&ampnbsp*</span></td>
                            </tr>
                            <tr>
                                <td colspan="2" id="msgName" style="color:#FF0000; font-family:Arial, Helvetica, sans-serif; font-size:10px;text-align:center"><?php echo $error_msg2;?></td>
                            </tr>
                            <tr><td colspan="2"><input type="hidden" id="hidProcess" name="hidProcess" /></td></tr>
                        </table>
                    </fieldset>
                </td>
            </tr>
        </table>
    </td>

```

```

</td>
</tr>
<tr height="15px"><td colspan="2"></td></tr>
<tr>
<td colspan="2">
<table cellpadding="0" cellspacing="0" align="center" border="0" width="95%">
<tr>
<td width="50%">
<fieldset style="width:50px">
<table cellpadding="0" cellspacing="0" align="center" border="0" width="80%" height="100%">
<tr>
<td><input type="button" value="Add" id="btnAdd" name="btnAdd" class="action_btn" onclick="btnAdd_Click()"></td>
<td>&nbsp;</td>
<td><input type="button" disabled="disabled" value="Modify" id="btnModify" name="btnModify" class="action_btn" onclick="btnModify_Click()"></td>
<td>&nbsp;</td>
<td><input type="button" disabled="disabled" value="Delete" id="btnDelete" name="btnDelete" class="action_btn" onclick="Confirm('Delete')"></td>
</tr>
</table>
</fieldset>
</td>
<td width="50%" align="right" style="padding-right:2px">
<fieldset style="width:50px">
<table cellpadding="0" cellspacing="0" align="center" border="0" width="80%" height="100%">
<tr>
<td><input type="submit" disabled="disabled" value="Save" id="btnSave" name="btnSave" class="action_btn"></td>
<td>&nbsp;</td>
<td><input type="button" disabled="disabled" value="Clear" id="btnClear" name="btnClear" class="action_btn" onclick="btnClear_Click()"></td>
<td>&nbsp;</td>
<td><input type="button" value="Exit" id="btnExit" name="btnExit" class="action_btn" onclick="Confirm('Exit')"></td>
</tr>
</table>
</fieldset>
</td>
</tr>
</table>
</td>
</tr>
<tr height="10px"><td colspan="2"></td></tr>
</table>
</td>
</form>

```

title.php: Title class with attributes and functions

<?php

```

class Title{
    var $titleCode;
    var $titleName;
    function Title($vals){
        $this->titleCode=$vals[0];
        $this->titleName=ucwords($vals[1]);
    }
    //function to save data
    function save(){
        $date= date("Y-m-d");
        $db_err_count=0;
        //Check whether the Title code is already exist.
        $select_query="SELECT * from title where titl_code='".$this->titleCode."'";
        $result_query=mysql_query($select_query);
        if(mysql_num_rows($result_query)>0){
            return "<font color='red'><b>$this->titleCode'</b> Code is Already Exist</font>";
        }
        else{
            $insert_query="insert into title(titl_code,titl_name)values('".$this->titleCode."','".$this->titleName')";
            if(!mysql_query($insert_query)){
                trigger_error("An Error Occured", E_USER_ERROR);
                $db_err_count++;
            }
            if(!mysql_query("commit;")){
                mysql_query("rollback");
                trigger_error("An Error Occured...", E_USER_ERROR);
                $db_err_count++;
            }
            else{
                if($db_err_count==0){
                    return "Record Added Successfully";
                }
            }
        }
    }
    //function to Update data
    function update(){
        $date= date("Y-m-d");
        $db_err_count=0;
        $update_query="UPDATE title set titl_name='".$this->titleName' where titl_code='".$this->titleCode."'";
        if(!mysql_query($update_query)){
            trigger_error("An Error Occured", E_USER_ERROR);
            $db_err_count++;
        }
        if(!mysql_query("commit;")){
            mysql_query("rollback");
            trigger_error("An Error Occured...", E_USER_ERROR);
            $db_err_count++;
        }
        else{
            if($db_err_count==0){
                return "Record Updated Successfully.";
            }
        }
    }
}

```

```

        }
    }
}

// function to Delete data
function remove(){
$db_err_count=0;
$delete_query="DELETE from title where titl_code='".$this->titleCode."'";
if(!mysql_query($delete_query)){
    trigger_error("An Error Occured", E_USER_ERROR);
    $db_err_count++;
}
if(!mysql_query("commit")){
    mysql_query("rollback");
    trigger_error("An Error Occured...", E_USER_ERROR);
    $db_err_count++;
}
else{
    if($db_err_count==0){
        return "Record Deleted Successfully.";
    }
}
}

}

?>

```

Master Data Module

Define Room Information

add_rooms.php : client side and web tier coding with HTML ,Java Script and PHP

```

<?php
ob_start();
session_start();
$v = $_SESSION['path'] = pathinfo($_SERVER['PHP_SELF']);
include("../includes/config.php");
include("../includes/errors.php");
include("../dbconn.php");
include("security.php");
$con=new ConnectionDB();

include("rooms.php");
include("functions/system_common.php");
$programme_name="Define Room";
$label1="Floor";
$label2="Room Type";
$label3="Room No";
$label4="Room Status";

if($_SERVER['REQUEST_METHOD'] == "POST"){
    $floorCode = trim($_REQUEST["floorCode"]);
    $rminRoomtyp = trim($_REQUEST["rminRoomtyp"]);
}

```

```

$rminRoomno = trim($_REQUEST["rminRoomno"]);
$status      = trim($_REQUEST["status"]);

$hidProcess = trim($_REQUEST["hidProcess"]);
$err_no=0;
if($floorCode==""){
    $err_no++;
    $error_msg1=validateMandatory($label1);
}
if($rminRoomtyp==""){
    $err_no++;
    $error_msg2=validateMandatory($label2);
}
if($rminRoomno==""){
    $err_no++;
    $error_msg3=validateMandatory($label3);
}
if($status==""){
    $err_no++;
    $error_msg4=validateMandatory($label4);
}

if($err_no==0){
//create an array to store values
$arrval=array($floorCode,$rminRoomtyp,$rminRoomno,$status);
$rom=new Rooms($arrval); //create an object from Room Class
//call save function
if($hidProcess=="add"){
    $msg=$rom->save();
}
//call modify function
if($hidProcess=="alter"){
    $msg=$rom->update();
}
//call delete function
if($hidProcess=="remove"){
    $msg=$rom->remove();
}
}
?>
<?php
include("../includes/grid.php");
?>
<script src="../js/validation.js"></script>
<script src="../js/ajax_script.js"></script>
<script language="javascript" type="text/javascript">
//function to enable text fields
function enableFileds(){
    document.getElementById("floorCode").disabled=false;
    document.getElementById("rminRoomtyp").disabled=false;
    document.getElementById("rminRoomno").disabled=false;
    document.getElementById("statusA").disabled=false;
    document.getElementById("statusD").disabled=false;
}

```

```

//function to disable text fields
    function disableFileds(){
        document.getElementById("floorCode").disabled=true;
        document.getElementById("rminRoomtyp").disabled=true;
        document.getElementById("rminRoomno").disabled=true;
        document.getElementById("statusA").disabled=true;
        document.getElementById("statusD").disabled=true;
    }

//function to Add button
    function btnAdd_Click(){
        clearAll();
        document.getElementById("btnAdd").disabled=true;
        document.getElementById("btnModify").disabled=true;
        document.getElementById("btnDelete").disabled=true;
        document.getElementById("btnSave").disabled=false;
        document.getElementById("btnClear").disabled=false;
        enableFileds();
        myObj1.disabled();
        document.getElementById("hidProcess").value="add";
    }

//function to clear text fields
    function btnClear_Click(){
        clearAll();
        document.getElementById("btnAdd").disabled=false;
        document.getElementById("btnSave").disabled=true;
        document.getElementById("btnClear").disabled=true;
        disableFileds();
        myObj1.enabled();
    }

    function clearAll(){
        document.getElementById("floorCode").value="";
        document.getElementById("rminRoomtyp").value="";
        document.getElementById("rminRoomno").value="";
    }

//function to modify button
    function btnModify_Click(){
        document.getElementById("hidProcess").value="alter";
        document.getElementById("rminRoomtyp").disabled=false;
        document.getElementById("statusA").disabled=false;
        document.getElementById("statusD").disabled=false;
        document.getElementById("btnSave").disabled=false;
        document.getElementById("btnClear").disabled=false;
        document.getElementById("btnAdd").disabled=true;
        document.getElementById("btnModify").disabled=true;
        document.getElementById("btnDelete").disabled=true;
    }

//Function to Exit & Delete Confirm
    function Confirm(para){
        if(para=="Exit"){
            var conf=confirm("Do you want to Exit?");
            if(conf==true){
                document.location='blank.php'
            }
        }
    }

```

```

        return true;
    }
    else{
        return false;
    }
}
if(para=="Delete"){
    var conf=confirm("Do you want to delete this record?");
    if(conf==true){
        enabledFields();
        document.getElementById("hidProcess").value="remove";
        document.frmRoominfo.submit();
        return true;
    }
    else{
        return false;
    }
}
}

</script>

<form name="frmRoominfo" method="post" action=<?php echo $PHP_SELF; ?><?php
echo $para?>" onsubmit="enableFileds()">
<td width=75% valign="top">
    <table cellpadding="0" cellspacing="0" border="0" width="96%" align="center">
        <tr>
            <td valign="top">
                <table cellpadding="0" cellspacing="0" align="center" border="0"
width="95%">
                    <tr height="5px"><td class="prgHeading"
bgcolor="#3573a6"><?php echo $programme_name?></td></tr>
                    <tr height="5px"><td></td></tr>
                    <tr height="10px"><td colspan="2" class="msg_td"><?php echo
$msg;?></td></tr>
                    <tr height="10px"><td colspan="2"></td></tr>
                </table>
            </td>
        </tr>
        <tr>
            <td valign="top">
                <table cellpadding="0" cellspacing="0" align="center" border="0"
width="100%" bgcolor="#FFFFFF">
                    <tr>
                        <td align="center">

```

<style>

```

#firstGrid .aw-column-0 {width: 150px;}
#firstGrid .aw-column-1 {width: 150px;}
#firstGrid .aw-column-2 {width: 150px;}
#firstGrid .aw-column-3 {width: 100px;}
#firstGrid .aw-grid-cell {border-right: 1px solid threelightshadow;}
#firstGrid .aw-grid-row {border-bottom: 1px solid threelightshadow;}
/* box model fix for strict doctypes, safari */
.aw-strict #myGrid .aw-grid-cell {padding-right: 3px;}

```

```

.aw-strict #myGrid .aw-grid-row {padding-bottom: 3px;}
```

```
</style>
```

```
<script>
<?php
```

```

$query_allrecords      = "select
floors.flor_name,roomtype.rmtyp_name,roominfo.rmin_numbr,roominfo.rmin_stat,roominfo.rm
in_floor,roominfo.rmin_rmtyp from roominfo,floors,roomtype where
floors.flor_code=roominfo.rmin_floor and roomtype.rmtyp_code=roominfo.rmin_rmtyp";
$result_allrecords    = mysql_query($query_allrecords);
$rows_allrecords      = mysql_num_rows($result_allrecords);
$query_select         = "select
floors.flor_name,roomtype.rmtyp_name,roominfo.rmin_numbr,roominfo.rmin_stat,roominfo.rm
in_floor,roominfo.rmin_rmtyp from roominfo,floors,roomtype where
floors.flor_code=roominfo.rmin_floor and roomtype.rmtyp_code=roominfo.rmin_rmtyp limit
10 offset 0";
$result_select        = mysql_query($query_select);
$rows_select          = mysql_num_rows($result_select);
$rows_select          = $rows_allrecords;
$record="";
while($record_select = mysql_fetch_array($result_select)){

$record.=
"[{$record_select[0]},{$record_select[1]},{$record_select[2]},{$record_select[3]},{$record_select[4]}
,$record_select[5]],";
}
$record.="";
?>
var myColumns = ["<?php echo $label1?>","<?php echo $label2?>","<?php echo
$label3?>","<?php echo $label4?>"];

var toolbarURL;
var gridLegend = 'Search by Column';
var gridSelect      = '--Select--';
var gridSelectMsg   ='Please select a search field.' ;
var searchBtn       = 'Search';
var lbltotalRecords = 'Total Records:';
var display         = 'Displaying :';

var gridAll      = 'All';
var param        = '<?php echo $query_allrecords?>';
var myData        = [<?php echo $record?>];
var str           = new AW.Formats.String;
var cellFormat = [str,str];
var Records      = <?php echo $rows_select?>;
var myObj1        = new AW.UI.Grid;
createGrid(myObj1
,'myObj1','firstGrid',myData,myColumns,1,cellFormat,Records,10,'handleGridAjaxRequests.p
hp',param);

myObj1.onSelectedRowsChanged = function(row){
document.getElementById("floorCode").value=this.getText(4,row);
document.getElementById("rminRoomtyp").value=this.getText(5,row);
document.getElementById("rminRoomno").value=this.getText(2,row);
```

```

var state=this.getCellText(3,row);
if(state=="A"){
    document.getElementById('statusA').checked=true;
}
if(state=="D"){
    document.getElementById('statusD').checked=true;
}
disableFileds();
document.getElementById("btnModify").disabled=false;
document.getElementById("btnDelete").disabled=false;
}
    </script>
</td>
</tr>
</table>
</td>
</tr>
<tr height="15px"><td colspan="2"></td></tr>
<tr>
    <td valign="top">
        <table cellpadding="0" cellspacing="0" align="center" border="0" width="100%" height="100%">
            <tr>
                <td align="center" valign="top">
                    <fieldset style="width:680">
                        <table cellpadding="0" cellspacing="0" align="center" width="60%" class="inputdata_tbl" border="0">
                            <tr height="6px"><td colspan="2"></td></tr>
                            <tr>
                                <td><?php echo $label1 ?></td>
                                <td>
                                    <select id="floorCode" disabled="disabled" name="floorCode" style="font-size:15px; font-family:Tw Cen MT" onBlur="validate('floorCode','msgFloor','<?php echo validateMandatory($label1)?>')">
                                        <option value="">[Please Select]</option>
                                        <?php
                                            $result_query=getFloor();
                                            while($record=mysql_fetch_array($result_query)){
                                                ?>
                                                <option value="<?php echo $record[0]?>"><?php if(trim($record[0])==trim($floorCode))echo "selected"; ?>><?php echo ucwords($record[1]);?></option>
                                            <?php } ?>
                                            </select><span class="req">&nbsp;*</span>
                                        </td>
                                    </tr>
                                    <tr>
                                        <td colspan="2" id="msgFloor" style="color:#FF0000; font-family:Arial, Helvetica, sans-serif; font-size:10px;text-align:center"><?php echo $error_msg1;?></td>
                                    </tr>
                                <tr height="8px"><td colspan="2"></td></tr>
                            </table>
                        </fieldset>
                    </td>
                </tr>
            </table>
        </td>
    </tr>
</table>

```

```

<tr>
    <td><?php echo $label2?></td>
    <td>
        <select id="rminRoomtyp" disabled="disabled" name="rminRoomtyp"
style="font-size:15px; font-family:Tw Cen MT"
onBlur="validate('rminRoomtyp','msgRmty','<?php echo validateMandatory($label2)?>')"
        <option value="">[Please Select]</option>

        <?php
            $result_query=getRoomType();
            while($record=mysql_fetch_array($result_query)){
                ?>
                <option value="<?php echo $record[0]?>"><?php if(trim
($record[0])==trim($rminRoomtyp))echo "selected"; ?>> <?php echo
ucwords($record[1]);?></option>

            <?php } ?>
            </select><span class="req">&nbsp;*</span>
        </td>
    </tr>
    <tr>
        <td colspan="2" id="msgRmty" style="color:#FF0000; font-family:Arial, Helvetica, sans-
serif; font-size:10px;text-align:center"><?php echo $error_msg2;?></td>
    </tr>
    <tr height="8px"><td colspan="2"></td></tr>
    <tr>
        <td><?php echo $label3?></td>
        <td><input type="text" style="width:115px" id="rminRoomno" name="rminRoomno"
maxlength="3" onkeyup="upperCase(this.id)" onchange="upperCase(this.id)" onkeypress="
return CharacNemariconly(event)" disabled="disabled" value="<?php echo $rminRoomno;?>"
onBlur="validate('rminRoomno','msgRmno','<?php echo
validateMandatory($label3)?>')"><span class="req">&nbsp;*</span></td>
    </tr>
    <tr>
        <td colspan="2" id="msgRmno" style="color:#FF0000; font-family:Arial, Helvetica,
sans-serif; font-size:10px;text-align:center"><?php echo $error_msg3;?></td>
    </tr>
    <tr height="8px"><td colspan="2"></td></tr>
    <tr>
        <td colspan="2" style="padding-left:150px"><input checked="checked"
disabled="disabled" type="radio" value="A" id="statusA"
name="status">&nbsp;Activate<input disabled="disabled" type="radio" id="statusD"
name="status" value="D" >&nbsp;Deactivate</td>
    </tr>
    <tr>
        <td colspan="2" style="color:#FF0000; font-family:Arial, Helvetica, sans-serif; font-
size:10px;text-align:center"></td>
    </tr>
    <tr><td colspan="2"><input type="hidden" id="hidProcess" name="hidProcess" /></td></tr>
</table>
</fieldset>

```

```

</td>
</tr>
</table>
</td>
</tr>
<tr height="15px"><td colspan="2"></td></tr>
<tr>
    <td colspan="2">
        <table cellpadding="0" cellspacing="0" align="center" border="0" width="100%" height="100%">
            <tr>
                <td width="50%">
                    <fieldset style="width:50px">
                        <table cellpadding="0" cellspacing="0" align="center" border="0" width="80%" height="100%">
                            <tr>
                                <td><input type="button" value="Add" id="btnAdd" name="btnAdd" class="action_btn" onclick="btnAdd_Click()"></td>
                                <td>&nbsp;</td>
                                <td><input type="button" disabled="disabled" value="Modify" id="btnModify" name="btnModify" class="action_btn" onclick="btnModify_Click()"></td>
                                <td>&nbsp;</td>
                                <td><input type="button" disabled="disabled" value="Delete" id="btnDelete" name="btnDelete" class="action_btn" onclick="Confirm('Delete')"></td>
                            </tr>
                        </table>
                    </fieldset>
                </td>
            </tr>
            <td width="50%" align="right">
                <fieldset style="width:50px">
                    <table cellpadding="0" cellspacing="0" align="center" border="0" width="80%" height="100%">
                        <tr>
                            <td><input type="submit" disabled="disabled" value="Save" id="btnSave" name="btnSave" class="action_btn"></td>
                            <td>&nbsp;</td>
                            <td><input type="button" disabled="disabled" value="Clear" id="btnClear" name="btnClear" class="action_btn" onclick="btnClear_Click()"></td>
                            <td>&nbsp;</td>
                            <td><input type="button" value="Exit" id="btnExit" name="btnExit" class="action_btn" onclick="Confirm('Exit')"></td>
                        </tr>
                    </table>
                </fieldset>
            </td>
        </tr>
    </table>
</td>
</tr>
</table>
</td>
</tr>
<tr height="10px"><td colspan="2"></td></tr>
</table>
</td>
</form>

```

rooms.php: Room class with attributes and functions

```
<?php
class Rooms{
    var $floorCode;
    var $rminRoomtyp;
    var $rminRoomno;
    var $status;
//constructor for class
    function Rooms($vals){
        $this->floorCode=$vals[0];
        $this->rminRoomtyp=$vals[1];
        $this->rminRoomno =$vals[2];
        $this->status = $vals[3];
    }
//function to insert new room
    function save(){
        $date= date("Y-m-d");
        $db_err_count=0;
//check whether the record is already exist.
        $select_query="SELECT * from roominfo where rmin_numbr='$this->rminRoomno'";
        $result_query=mysql_query($select_query);
        if(mysql_num_rows($result_query)>0){
            return "<font color='red'>Record is Already Exist</font>";
        }
        else{
            $insert_query="insert into roominfo(rmin_floor,rmin_rmtyp, rmin_numbr,
rmin_stat)values('$this->floorCode','$this->rminRoomtyp','$this->rminRoomno','$this-
>status')";
            if(!mysql_query($insert_query)){
                trigger_error("An Error Occured", E_USER_ERROR);
                $db_err_count++;
            }
            if(!mysql_query("commit")){
                mysql_query("rollback");
                trigger_error("An Error Occured...", E_USER_ERROR);
                $db_err_count++;
            }
            else{
                if($db_err_count==0){
                    return "Record Added Successfully";
                }
            }
        }
    }
//function to update room details.
    function update(){
        $date= date("Y-m-d");
        $db_err_count=0;
        $update_query="UPDATE roominfo set rmin_stat='$this-
>status',rmin_rmtyp='$this->rminRoomtyp' where rmin_numbr='$this->rminRoomno'";
        if(!mysql_query($update_query)){
            trigger_error("An Error Occured", E_USER_ERROR);
            $db_err_count++;
        }
    }
}
```

```

        }
        if(!mysql_query("commit;")){
            mysql_query("rollback");
            trigger_error("An Error Occured...", E_USER_ERROR);
            $db_err_count++;
        }
        else{
            if($db_err_count==0){
                return "Record Updated Successfully";
            }
        }
    }

// function to delete room details.
function remove(){
$db_err_count=0;
$delete_query="DELETE from roominfo where rmin_numbr='$this->rminRoomno'";
if(!mysql_query($delete_query)){
    trigger_error("An Error Occured", E_USER_ERROR);
    $db_err_count++;
}
if(!mysql_query("commit;")){
    mysql_query("rollback");
    trigger_error("An Error Occured...", E_USER_ERROR);
    $db_err_count++;
}
else{
    if($db_err_count==0){
        return "Record Deleted Successfully";
    }
}
}
?>

```

Company Profile Module

Profile Information

add_comprofile.php: client side and web tier coding with HTML, Java Script and PHP

```

<?php

ob_start();
session_start();
$v = $_SESSION['path'] = pathinfo($_SERVER['PHP_SELF']);
include("../includes/config.php");
include("../includes/errors.php");
include("../dbconn.php");
include("security.php");
$con=new ConnectionDB();
include("functions/system_common.php");

```

```

include("comprofile.php");
$label1="Company Name";
$label2="Street";
$label3="Street 2";
$label4="City";
$label5="Country";
$label6="Primary Phone No";
$label7="Secondary Phone No";
$label8="Fax";
$label9="E-mail";
$label10="Company Logo";

$programme_name="Company Profile";
if($_SERVER['REQUEST_METHOD'] == "POST"){

    $comName      = trim($_REQUEST["comName"]);
    $comStree1    = trim($_REQUEST["comStree1"]);
    $comStree2   = trim($_REQUEST["comStree2"]);
    $comCity     = trim($_REQUEST["comCity"]);
    $comCuntry   = trim($_REQUEST["comCuntry"]);
    $comPno      = trim($_REQUEST["comPno"]);
    $comSno      = trim($_REQUEST["comSno"]);
    $comFax      = trim($_REQUEST["comFax"]);
    $comEmail    = trim($_REQUEST["comEmail"]);
    $hidProcess  = trim($_REQUEST["hidProcess"]);
    $err_no=0; //if error is occurred check where it is

    if($comName==""){
        $err_no++;
        $error_msg1=validateMandatory($label1);
    }
    if($comStree1==""){
        $err_no++;
        $error_msg2=validateMandatory($label2);
    }
    if($comCity==""){
        $err_no++;
        $error_msg3=validateMandatory($label4);
    }
    if($comCuntry==""){
        $err_no++;
        $error_msg4=validateMandatory($label5);
    }
    if($comPno==""){
        $err_no++;
        $error_msg5=validateMandatory($label6);
    }
    if($image==""){
        $err_no++;
        $error_msg6=validateMandatory($label10);
    }
    if(!$image){
        $err_no++;
    }
}

```

```

        $error_msg6=validateImage();
    }
    if($image){
        $filename = stripslashes($_FILES['image']['name']);
        $extension = getExtension($filename);
        $extension = strtolower($extension);
        if (($extension != "jpg") && ($extension != "jpeg") && ($extension != "png")
&& ($extension != "gif")){
            $error_msg6=checkExtension();
            $err_no++;
        }
        else{
/*           $size=filesize($_FILES['image']['tmp_name']);
           if ($size > MAX_SIZE*1024*1024){
               $error_msg6=checkSizeOfFile();
               $err_no++;
           }*/
           $image_name=time().'.'.$extension;
           $newname="images/".$image_name;
           $copied = copy($_FILES['image']['tmp_name'], $newname);
           if (!$copied) {
               $error_msg6=checkIsfileCopy();
               $err_no++;
           }
       }
   }

   if($err_no==0){
       $arrval=array($comName,$comStree1,$comStree2,$comCity,$comCuntry,$comPno,$
comSno,$comFax,$comEmail,$newname);
       $com= new Comprofile($arrval);//create an object from Comprofile
       if($hidProcess=="add"){//Call Add Function
           $msg=$com->save();
       }
   }
?>
<?php
  include("../includes/grid.php");
?>

<form name="frmCompany" method="post" action="<?php echo $PHP_SELF; ?>?<?php
echo $para?>" enctype="multipart/form-data">

<?php

$select_company="select
comp_name,comp_str1,comp_str2,comp_city,comp_cuntry,comp_ppno,comp_spno,comp_fax,
comp_email,comp_logo from comprofile";
$result_query=mysql_query($select_company);
if(mysql_num_rows($result_query)>0)
{
$record=mysql_fetch_array($result_query);

```

```

        $comName=$record[0];
        $comStree1=$record[1];
        $comStree2=$record[2];
        $comCity=$record[3];
        $comCuntry=$record[4];
        $comPno=$record[5];
        $comSno=$record[6];
        $comFax=$record[7];
        $comEmail=$record[8];
        $comLogo=$record[9];

    }

?>
<td width=75%>
    <table cellpadding="0" cellspacing="0" border="0" width="96%" height="100%" align="left">
        <tr>
            <td valign="top">
                <table cellpadding="0" cellspacing="0" align="center" border="0" width="95%">
                    <tr height="5px"><td class="prgHeading" bgcolor="#3573a6"><?php echo $programme_name?></td></tr>
                    <tr height="5px"><td></td></tr>
                    <tr height="10px"><td colspan="2" class="msg_td"><?php echo $msg;?></td></tr>
                    <tr height="10px"><td colspan="2"></td></tr>
                </table>
            </td>
        </tr>
        <tr>
            <td valign="top">
                <table cellpadding="0" cellspacing="0" align="center" border="0" width="80%" height="300px">
                    <tr>
                        <td>
                            <fieldset>
                                <table cellpadding="0" cellspacing="0" align="center" border="0" width="100%" height="400px" class="company_tbl">
                                    <tr>
                                        <td><?php echo $label1?></td>
                                        <td><input type="text" size="40px" name="comName" id="comName" maxlength="40" value="<?php echo $comName?>" onBlur="validate('comName','msgcomname','<?php echo validateMandatory($label1)?>')"/><span class="req">&nbsp;*</span></td>
                                    </tr>
                                    <tr>
                                        <td colspan="2" id="msgcomname" style="color:#FF0000; font-family:Arial, Helvetica, sans-serif; font-size:10px;text-align:center"><?php echo $error_msg1;?></td>
                                    </tr>
                                    <tr>
                                        <td><?php echo $label2?></td>
                                        <td><input type="text" size="30px" nae="comStree1" id="comStree1" maxlength="60" value="<?php echo $comStree1?>">
                                    </tr>
                                </table>
                            </fieldset>
                        </td>
                    </tr>
                </table>
            </td>
        </tr>
    </table>

```

```

onBlur="validate('comStreet1','msgstreet1','<?php echo
validateMandatory($label2)?>')"/><span class="req">&nbsp;*</span></td>
</tr>
<tr>
    <td colspan="2" id="msgstreet1" style="color:#FF0000; font-
family:Arial, Helvetica, sans-serif; font-size:10px;text-align:center"><?php echo
$error_msg2;?></td>
</tr>
<tr>
    <td><?php echo $label4?></td>
    <td><input type="text" size="30px" name="comStreet2" id="comStreet2" maxlength="60" value="<?php echo $comStreet2?>"/></td>
</tr>
<tr>
    <td colspan="2" id="msgcity" style="color:#FF0000; font-
family:Arial, Helvetica, sans-serif; font-size:10px;text-align:center"><?php echo
$error_msg3;?></td>
</tr>
<tr>
    <td><?php echo $label5?></td>
    <td><select id="comCuntry" name="comCuntry" style="font-
size:15px; font-family:Tw Cen MT" onBlur="validate('comCuntry','msgcoun','<?php echo
validateMandatory($label5)?>')">
        <option value="">[Please Select]</option>
        <?php
            $query_select="SELECT coun_code,coun_name from country";
            $result_query=mysql_query($query_select);
            while($record=mysql_fetch_array($result_query)){
        ?>
        <option value="<?php echo $record[0]?>" <?php if(trim
($record[0])==trim($comCuntry))echo "selected"; ?>><?php echo
ucwords($record[1]);?></option>
        <?php } ?>
    </select><span class="req">&nbsp;*</span>
    </td>
</tr>
<tr>
    <td colspan="2" id="msgcoun" style="color:#FF0000; font-
family:Arial, Helvetica, sans-serif; font-size:10px;text-align:center"><?php echo
$error_msg4;?></td>
</tr>
<tr>
    <td><?php echo $label6?></td>
    <td><input type="text" size="30px" name="comPno" id="comPno" maxlength="20" value="<?php echo $comPno?>">
    <?php echo validate('comPno','msgppno','<?php echo validateMandatory($label5)?>')?><span
class="req">&nbsp;*</span></td>
</tr>

```

```

<tr>
    <td colspan="2" id="msgppno" style="color:#FF0000; font-
family:Arial, Helvetica, sans-serif; font-size:10px;text-align:center"><?php echo
$error_msg5;?></td>
</tr>
<tr>
    <td><?php echo $label7?></td>
    <td><input type="text" size="30px" name="comSno" id="comSno"
maxlength="20" value="<?php echo $comSno?>"></td>
</tr>
<tr>
    <td><?php echo $label8?></td>
    <td><input type="text" size="30px" name="comFax" id="comFax"
maxlength="20" value="<?php echo $comFax?>"></td>
</tr>
<tr>
    <td><?php echo $label9?></td>
    <td><input type="text" size="30px" name="comEmail"
id="comEmail" maxlength="40" value="<?php echo $comEmail?>"></td>
</tr>
<tr>
    <td valign="top"><?php echo $label10?></td>
    <td><?php if($comLogo==""){ ?><?php }else{ ?><?php }?></td>
</tr>
<tr>
    <td></td>
    <td><input type="file" name="image"
onBlur="validate('image','msgimg','<?php echo validateMandatory($label6)?>')"></td>
</tr>
<tr>
    <td colspan="2" id="msgimg" style="color:#FF0000; font-
family:Arial, Helvetica, sans-serif; font-size:10px;text-align:center"><?php echo
$error_msg6;?></td>
</tr>
<tr><td colspan="2"><input type="hidden" id="hidProcess"
name="hidProcess" value="add"/></td></tr>
</table>
</fieldset>
</td>
</tr>
<tr height="15px"><td></td></tr>
<tr>
    <td width="50%" align="right">
    <fieldset style="width:50px">
        <table cellpadding="0" cellspacing="0" align="center" border="0"
width="80%" height="100%">
            <tr>
                <td><input type="submit" value="Save" id="btnSave"
name="btnSave" class="action_btn"></td>
                <td>&nbsp;</td>
                <td><input type="button" disabled="disabled" value="Clear"
id="btnClear" name="btnClear" class="action_btn" onclick="btnClear_Click()"/></td>
                <td>&nbsp;</td>
            
```

```

        <td><input type="button" value="Exit" id="btnExit" name="btnExit"
class="action_btn" onclick="Confirm('Exit')"></td>
    </tr>
    </table>
    </fieldset>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</form>

```

comprofile.php: Company Profile class with attributes and functions

```
<?php
```

```

class Comprofile{
    var $comName;
    var $comStreet;
    var $comStreet2;
    var $comCity;
    var $comCountry;
    var $comPno;
    var $comSno;
    var $comFax;
    var $comEmail;
    var $newname;
    function Comprofile($vals){
        $this->comName = $vals[0];
        $this->comStreet = $vals[1];
        $this->comStreet2 = $vals[2];
        $this->comCity = $vals[3];
        $this->comCountry = $vals[4];
        $this->comPno = $vals[5];
        $this->comSno = $vals[6];
        $this->comFax = $vals[7];
        $this->comEmail = $vals[8];
        $this->newname = $vals[9];
    }
    function save(){
        $db_err_count = 0;
        $date = date("Y-m-d");
        $insert_comprofile = "insert into
comprofile(comp_name,comp_str1,comp_str2,comp_city,comp_country,comp_ppno,comp_spn
o,comp_fax,comp_email,comp_logo) values('$this->comName','$this->comStreet','$this-
>comStreet2','$this->comCity','$this->comCountry','$this->comPno','$this->comSno','$this-
>comFax','$this->comEmail','$this->newname')";
        if(!mysql_query($insert_comprofile)){
            trigger_error("An Error Occured", E_USER_ERROR);
            $db_err_count++;
        }
        if(!mysql_query("commit")){

```

```

        mysql_query("rollback");
        trigger_error("An Error Occured...", E_USER_ERROR);
        $db_err_count++;
    }
    else{
        if($db_err_count==0){
            return "Record Added Successfully";
        }
    }
}

function update(){
    $db_err_count=0;
    $date= date("Y-m-d");
    $update_comprofile="update comprofile set comp_name='".$this->comName',comp_str1='".$this->comStree1',comp_str2='".$this->comStree2',comp_city='".$this->comCity"',comp_cuntry='".$this->comCuntry',comp_ppno='".$this->comPno',comp_spno='".$this->comSno',comp_fax='".$this->comFax',comp_email='".$this->comEmail',comp_logo='".$this->newname."'";
    if(!mysql_query($update_comprofile)){
        trigger_error("An Error Occured", E_USER_ERROR);
        $db_err_count++;
    }
    if(!mysql_query("commit;")){
        mysql_query("rollback");
        trigger_error("An Error Occured...", E_USER_ERROR);
        $db_err_count++;
    }
    else{
        if($db_err_count==0){
            return "Record Updated Successfully";
        }
    }
}
}

?>

```

Access and Security Control Module

Group User

create_users.php: client side and web tier coding with HTML ,Java Script and PHP

```

<?php
ob_start();
session_start();
$v = $_SESSION['path'] = pathinfo($_SERVER['PHP_SELF']);
include("../includes/config.php");
include("../includes/errors.php");
include("../dbconn.php");
include("security.php");

```

```

$con=new ConnectionDB();
include("createusers.php");
include("functions/system_common.php");

$programme_name="Create Users";
$label1="User Group Code";
$label2="All Users";
$label3="Assigned Users";
$label4="User Name";
$label5="Employee No";
$label6="Title";
$label7="Full Name";
$label8="NIC No";
$label9="Designation";
$label10="Date of Join";
$label11="Photo";
if($_SERVER['REQUEST_METHOD'] == "POST"){
    $usrGrp = trim($_REQUEST["usrGrp"]);
    $hidRefid = trim($_REQUEST["hidRefid"]);
    $err_no =0;
    //Check empty fields
    if($usrGrp==""){
        $err_no++;
        $error_msg1=validateMandatory($label1);
    }
    if($err_no==0){
        $arrval=array($usrGrp,$hidRefid);
        $crt=new CreateUsers($arrval);//create an object from Create User Class
        $msg=$crt->saveCreateUsers();
    }
}

?>
<?php
    include("../includes/grid.php");
?>

<script src="../js/doublegrid.js"></script>
<script src="../js/ajax_script.js"></script>

<script type="text/javascript">
function get_value(){
    var opval="";
    for (var i=0;i<document.frmCreateUser.mapUser.options.length;i++ ) {
        opval+=document.frmCreateUser.mapUser.options[i].value+",";
    }
    document.getElementById("hidRefid").value=opval;
}
function getUserNotAssigned(){
    document.getElementById("unmapUser").disabled=false;
    document.getElementById("mapUser").disabled=false;
    document.getElementById("btn_next").disabled=false;
    document.getElementById("btn_nextall").disabled=false;
    document.getElementById("btn_previous").disabled=false;
    document.getElementById("btn_previousall").disabled=false;
}

```

```

        document.getElementById("btn_next").className="td_nextbg";
        document.getElementById("btn_nextall").className='td_nextallbg';
        document.getElementById("btn_previous").className='td_previousbg';
        document.getElementById("btn_previousall").className='td_previousallbg';
        var Source = document.getElementById("usrGrp");
        var val = Source.options[Source.options.selectedIndex].value;
        var XMLHttpRequestObj=getNewXmlHttpRequest();
        var file="ajax_result.php";
        var param="action=getNotAsgUsr&ugrp="+val;
        var callbackHandler = getReadyStateHandler(xmlHttpRequestObj,getValues);
        XMLHttpRequestObj.onreadystatechange=callbackHandler;
        XMLHttpRequestObj.open("POST",file,true);
        XMLHttpRequestObj.setRequestHeader('Content-Type','application/x-www-form-urlencoded');
        XMLHttpRequestObj.send(param);
    }
    function getValues(responseText){
        msg=responseText.split("/");
        var roleSelector = document.getElementById("unmapUser");
        while(roleSelector.hasChildNodes()){
            roleSelector.removeChild(roleSelector.lastChild);
        }
        for(var i=0; i<msg.length-1;i++){
            var txtmsg=msg[i].split(",");
            for(var x=0; x<txtmsg.length-1;x++){
                var newOptionNode = document.createElement("option");
                var textNode =document.createTextNode(txtmsg[1]);
                newOptionNode.appendChild(textNode);
                newOptionNode.setAttribute("value", txtmsg[0]);
                roleSelector.appendChild(newOptionNode);
            }
        }
    }
    function getUserAssigned(){
        document.getElementById("unmapUser").disabled=false;
        document.getElementById("mapUser").disabled=false;
        document.getElementById("btn_next").disabled=false;
        document.getElementById("btn_nextall").disabled=false;
        document.getElementById("btn_previous").disabled=false;
        document.getElementById("btn_previousall").disabled=false;
        document.getElementById("btn_next").className="td_nextbg";
        document.getElementById("btn_nextall").className='td_nextallbg';
        document.getElementById("btn_previous").className='td_previousbg';
        document.getElementById("btn_previousall").className='td_previousallbg';
        var Source = document.getElementById("usrGrp");
        var val = Source.options[Source.options.selectedIndex].value;
        var XMLHttpRequestObj=getNewXmlHttpRequest();
        var file="ajax_result.php";
        var param="action=getAsgUsr&ugrp="+val;
        var callbackHandler = getReadyStateHandler(xmlHttpRequestObj,getMapValues);
        XMLHttpRequestObj.onreadystatechange=callbackHandler;
        XMLHttpRequestObj.open("POST",file,true);
        XMLHttpRequestObj.setRequestHeader('Content-Type','application/x-www-form-urlencoded');
        XMLHttpRequestObj.send(param);
    }
}

```

```

function getMapValues(responseText){
    msg=responseText.split("/");
    var roleSelector = document.getElementById("mapUser");
    while(roleSelector.hasChildNodes()){
        roleSelector.removeChild(roleSelector.lastChild);
    }
    for(var i=0; i<msg.length-1;i++){
        var txtmsg=msg[i].split(",");
        for(var x=0; x<txtmsg.length-1;x++){
            var newOptionNode = document.createElement("option");
            var textNode =document.createTextNode(txtmsg[1]);
            newOptionNode.appendChild(textNode);
            newOptionNode.setAttribute("value", txtmsg[0]);
            roleSelector.appendChild(newOptionNode);
        }
    }
}

function clearAll(){
    document.getElementById("usrGrp").value="";
    var roleSelector = document.getElementById("mapUser");
    while(roleSelector.hasChildNodes()){
        roleSelector.removeChild(roleSelector.lastChild);
    }
}

function Confirm(para){
    if(para=="Exit"){
        var conf=confirm("Do you want to Exit?");
        if(conf==true){
            document.location='index.php'
            return true;
        }
        else{
            return false;
        }
    }
}

function getCode(val){
    var Source =val;
    var val = Source.options[Source.options.selectedIndex].value;
    var val2 = Source.options[Source.options.selectedIndex].text;
    document.getElementById("commonname").value=val2;
    var xmlhttpObj=getNewXmlHttpRequest();
    var file="ajax_result.php";
    var param="action=udrdetails&refCode="+val;
    var callbackHandler = getReadyStateHandler(xmlhttpObj,getUsers);
    xmlhttpObj.onreadystatechange=callbackHandler;
    xmlhttpObj.open("POST",file,true);
    xmlhttpObj.setRequestHeader('Content-Type','application/x-www-form-urlencoded');
    xmlhttpObj.send(param);
}

function getUsers(responseText){

```

```

msg=responseText.split("@@@");
document.getElementById("empNo").value=msg[0];
document.getElementById("empTitle").value=msg[1];
document.getElementById("fullname").value=msg[2];
document.getElementById("empNid").value=msg[3];
document.getElementById("desigCode").value=msg[4];
document.getElementById("empDoj").value=msg[5];
if(msg[6]!=""){
    document.frmCreateUser.photo.src=msg[6];
}
else{
    document.frmCreateUser.photo.src="../../images/default.jpg";
}

}
</script>

<form name="frmCreateUser" method="post" action=<?php echo $PHP_SELF; ?><?php
echo $para?>">
<td width="75%" valign="top">
    <table cellpadding="0" cellspacing="0" border="0" width="96%" align="left">
        <tr>
            <td valign="top">
                <table cellpadding="0" cellspacing="0" align="center" border="0"
width="95%">
                    <tr height="5px"><td class="prgHeading"
bgcolor="#3573a6"><?php echo $programme_name?></td></tr>
                    <tr height="5px"><td></td></tr>
                    <tr height="10px"><td colspan="2" class="msg_td"><?php echo
$msg;?></td></tr>
                    <tr height="10px"><td colspan="2"></td></tr>
                </table>
            </td>
        </tr>
        <tr>
            <td>
                <table width="100%" cellspacing="0" cellpadding="3"
border="0" class="inputdata_tbl">
                    <tr>
                        <td style="padding-left:20px">
                            <table cellpadding="0" cellspacing="0" align="center" border="0"
width="100%" class="inputdata_tbl">
                                <tr>
                                    <td width="40px"><?php echo $label1?></td>
                                    <td width="170px">
                                        <select id="usrGrp" name="usrGrp" style="font-size:15px;
font-family:Tw Cen MT; width:150px"
onchange="getUserNotAssigned();getUserAssigned()">
                                            <option value="">[Please Select]</option>
                                            <?php
                                                $result_query=getUserGrp();
                                                while($record=mysql_fetch_array($result_query)){
                                            ?>

```

```

        <option value="<?php echo $record[0]?>"><?php if(trim
($record[0])==trim($usrGrp))echo "selected"; ?> > <?php echo
ucwords($record[1]);?></option>
            <?php } ?>
            </select><span class="req">&nbsp;*</span>
        </td>
    </tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
<tr height="10px"><td colspan="2"></td></tr>
<td valign="top" style="padding-left:20px">
    <table cellpadding="0" cellspacing="0" align="center" border="0"
width="100%">
        <tr>
            <td width="44%">
                <fieldset>
                    <legend style="font-family:Arial;font-size:12px;color:#195579; font-
weight:bold"><?php echo $label2?></legend>
                    <table cellpadding="0" cellspacing="0" align="center" border="0"
width="100%">
                        <tr>
                            <td>
                                <select disabled="disabled" name="unmapUser" id="unmapUser"
style="width:320;height:150px;font-size:11px; font-family:Arial" size="5"
multiple="multiple" onChange="getCode(this);">
                                    <?php
                                    $result_query=getUserNotAsg($usrGrp);
                                    while($record=mysql_fetch_array($result_query)){
                                    ?>

                                    <option value="<?php echo $record[0]?>"><?php if(trim
($record[0])==trim($unmapUser))echo "selected"; ?> > <?php echo $record[1];?></option>
                                    <?php }?>
                                </select>
                            </td>
                        </tr>
                    </table>
                </fieldset>
            </td>
        <td width="10%">
            <table align="center" cellpadding="0" cellspacing="0" border="0" width="100%">
                <tr height="20px"><td></td></tr>
                <tr height="25px">
                    <td>
                        <table align="center" cellpadding="0" cellspacing="0" border="0"
width="30px">
                            <tr height="30px">
                                <td width="26px"><input type="button"
onClick="MoveItem('unmapUser', 'mapUser');" class="td_disnextbg" disabled="disabled"
id="btn_next"></td>

```

```

        </tr>
        </table>
        </td>
        </tr>
        <tr height="30px">
            <td>
                <table align="center" cellpadding="0" cellspacing="0" border="0"
width="30px">
                    <tr height="23px">
                        <td width="26px"><input type="button"
onclick="MoveAllItem('unmapUser', 'mapUser');" class="td_disnextallbg"
disabled="disabled" id="btn_nextall"></td>
                    </tr>
                    </table>
                </td>
            </tr>
            <tr height="30px">
                <td>
                    <table align="center" cellpadding="0" cellspacing="0" border="0"
width="30px">
                        <tr height="23px">
                            <td width="26px"><input type="button"
onclick="MoveItem('mapUser', 'unmapUser');" class="td_dispreviousbg" disabled="disabled"
id="btn_previous"></td>
                        </tr>
                        </table>
                    </td>
                </tr>
                <tr height="30px">
                    <td>
                        <table align="center" cellpadding="0" cellspacing="0" border="0"
width="30px">
                            <tr height="23px">
                                <td width="26px"><input type="button" onClick="MoveAllItem('mapUser',
'unmapUser');" class="td_dispreviousallbg" disabled="disabled" id="btn_previousall"></td>
                            </tr>
                        </table>
                    </td>
                </tr>
                <tr height="30px">
                    <td>
                </td>
            </tr>
            </table>
        </td>
    </tr>
    </table>
</td>
<td width="44%" >
<fieldset>
    <legend style="font-family:Arial;font-size:12px;color:#195579; font-weight:bold"><?php echo $label3?></legend>
    <table align="center" cellpadding="0" cellspacing="0" border="0">
        <tr>
            <td>
                <select disabled="disabled" name="mapUser" id="mapUser"
style="width:320;height:150px;font-size:11px; font-family:Arial" size="5"
multiple="multiple" onChange="getCode(this);">
                    <?php
                        $result_query=getUserAsg($usrGrp);
                        while($record=mysql_fetch_array($result_query)){
                    ?>

```

```

<option value=<?php echo $record[0]?><?php if(trim ($record[0])==trim($mapUser))echo "selected"; ?>><?php echo $record[2];?></option>
        <?php }?>
    </select>
</td>
</tr>
</table>
</fieldset>
</td>
</tr>
<tr><td><input type="hidden" id="hidRefid" name="hidRefid"> </td></tr>
</table>
</td>
</tr>
<tr height="10px"><td>&nbsp;</td></tr>
<tr>
    <td colspan="2" style="padding-left:6px">
        <table cellpadding="0" cellspacing="0" align="center" border="0" width="96%" height="100%">
            <tr>
                <td width="60%">
                    <fieldset>
                        <legend style="font-family:Arial;font-size:12px;color:#195579; font-weight:bold">User Information</legend>
                        <table cellpadding="0" cellspacing="0" align="center" border="0" width="95%" class="inputdata_tbl">
                            <tr height="10px"><td></td></tr>
                            <tr>
                                <td><?php echo $label4?></td>
                                <td> <input disabled name="commonname" type="text" size="20" ID="commonname"></td>
                            </tr>
                            <tr height="5px"><td></td></tr>
                            <tr> <td><?php echo $label5?></td>
                                <td> <input disabled name="empNo" type="text" size="20" ID="empNo"></td>
                            </tr>
                            <tr height="5px"><td></td></tr>
                            <tr>
                                <td><?php echo $label6?></td>
                                <td> <input disabled name="empTitle" type="text" size="20" ID="empTitle"></td>
                            </tr>
                            <tr height="5px"><td></td></tr>
                            <tr>
                                <td><?php echo $label7?></td>
                                <td><input disabled="disabled" name="fullname" type="text" size="40" ID="fullname"></td>
                            </tr>
                            <tr height="5px"><td></td></tr>
                            <tr>
                                <td><?php echo $label8?></td>
                                <td ><input disabled="disabled" type="text" style="width:120px" name="empNid" id="empNid" maxlength="20"/></td>
                            </tr>

```

```

<tr height="5px"><td></td></tr>
<tr>
    <td><?php echo $label9?></td>
    <td><input disabled name="desigCode" type="text" size="20" id="desigCode"></td>
</tr>
<tr height="5px"><td></td></tr>
<tr>
    <td><?php echo $label10?></td>
    <td><input disabled="disabled" type="text" style="width:120px" name="empDoj" id="empDoj" maxlength="20" /></td>
</tr>
<tr height="5px"><td></td></tr>
</table>
</fieldset>
</td>
<td width="2%">&ampnbsp</td>
<td width="38%" valign="top">
<fieldset>
<legend style="font-family:Arial;font-size:12px;color:#195579; font-weight:bold"><?php echo $label11?></legend>
    <table cellpadding="0" cellspacing="0" align="center" border="0" width="80%" height="100%" class="inputdata_tbl">
        <tr>
            <td width="48%">
                <table width="100%" border="0" cellspacing="0" cellpadding="0">
                    <tr>
                        <td align="center"></td>
                    </tr>
                </table>
            </td>
        </tr>
    </table>
</td>
</tr>
</table>
</fieldset>
</td>
</tr>
</table>
</td>
</tr>
<tr>
<td colspan="2">
    <table cellpadding="0" cellspacing="0" align="center" border="0" width="100%" height="100%">
        <tr>
            <td width="50%" align="right" style="padding-left:80px">
                <fieldset style="width:50px">
                    <table cellpadding="0" cellspacing="0" align="center" border="0" width="80%" height="100%">
                        <tr height="10px"><td></td></tr>
                        <tr>
                            <td><input type="submit" value="Save" id="btnSave" name="btnSave" class="action_btn" onclick="get_value()"></td>
                            <td>&ampnbsp</td>
                        </tr>
                    </table>
                </fieldset>
            </td>
        </tr>
    </table>
</td>
</tr>

```

```

<td><input type="button" value="Clear" id="btnClear" name="btnClear"
class="action_btn" onClick="clearAll()"></td>
<td>&nbsp;</td>
<td><input type="button" value="Exit" id="btnExit" name="btnExit"
class="action_btn" onClick="Confirm('Exit')"></td>
</tr>
<tr height="6px"><td></td></tr>
</table>
</fieldset>
</td>
</tr>
</table>
</td>
</tr>
<tr height="10px"><td colspan="2"></td></tr>
</table>
</td>
</form>

```

createusers.php: Create Users class with attributes and functions

```

<?php
class CreateUsers{
    var $usrGrp;
    var $hidRefid;

    function CreateUsers($vals){
        $this->usrGrp=$vals[0];
        $this->hidRefid=$vals[1];
    }
    //function add created users
    function saveCreateUsers(){
        $delete_query="DELETE from grpusers where gru_acsgrp='$this->usrGrp'";
        $result_query=mysql_query($delete_query);
        $Name=explode(",",$this->hidRefid);
        $count=sizeof($Name);
        for($i=0;$i<$count-1;$i++){
            $query_insert="insert into grpusers(gru_acsgrp,gru_usrref)values('$this-
>usrGrp','$Name[$i]')";
            $result_query=mysql_query($query_insert);
        }
        return "Record Added Successfully";
    }
}
?>

```

Appendix G - Client Certificate

The approval certification issued by Mr. Prasanna, the Manager of the client or organization – Dream Beach Resort, is included in the next page.



Sep 5th 2010

BIT Coordinator
External Degrees Centre of UCSC,
221/2A, Dharmapala Mawatha,
Colombo 07.

Dear Sir,

APPROVING WEB BASED HOTEL MANAGEMENT SYSTEM

As per the request made by P.S.S. Swarnamali (Registration no: R051992) who is a student of BIT external degree programme conducted by the University of Colombo School of Computing, Reid Avenue, Colombo-03, she was granted permission to convert the current manual system into a web based system titled as 'Web based Hotel Management System' as her final year project.

I wish to inform that the candidate has successfully completed the proposed solution by fulfilling the requirements of the company. The overall system has been evaluated by the company staff and it has been identified as the most suitable solution to manage operation of the company in an effective manner.

The company would like to thank Miss. P.S.S. Swarnamali for this accomplishment while appreciating the effort that has been taken to automate the company current system. We wish her every success in all his future endeavours.

Yours sincerely

W.G.Prasanna Chandimal
.....
Manager
Dream Beach Resort
Manager
Ampalangoda
Dream Beach Resort

06/11/2010

Date

Glossary and Index

A	Activity: Behavior that occurs while in a state.
	Actor: Someone or something external to the system that must interact with the system under development.
	Apache: Apache, otherwise known as Apache HTTP Server , is an established standard in the online distribution of website services, which gave the initial boost for the expansion of the World Wide Web.
	Attribute: A data definition held by objects of a class.
B	Back end:
C	Class: A description of a group of objects with common properties, common behavior, common relationships to other objects.
D	Data: Facts or observations about actual events or business transactions. They are objective measurements of the attributes of entities such as people, places, things and events.
	Data Grid: computing system that primarily deals with large data sets. They are used for the controlled sharing and management of large amounts of distributed data. They are closely related to computational grid computing systems.
	Database: Collection of logically related records or file
E	Entity: A single object about which data can be stored. It is the ‘subject’ of a table.
F	Functional Requirements: Services provided by the system to satisfy the client’s expectations
H	Hotel Management System: That helps manage reservations, bookings, roomstays, room planning, guests, accounts, folios, agents, pricing plans, basically front office and back office hotel operations.
	HTML: The common scripting language for interchange of hypertext between the world wide web client and server. Web pages must be written using HTML.
	HTTP: Hypertext Transfer Protocol: A protocol commonly used to access resources on the Internet.
I	Income Analysis: The process of performing the income approach or a summary of the research and calculations performed during the income approach to value real estate.
	Information: Data placed in a meaningful and useful context for an end user.

	Internet: A worldwide network of computers that can be accessed via the university computer network.
J	JavaScript: A scripting language used in web content development, primarily to create functions that can be embedded in or included from HTML (Hypertext Markup Language) documents and that interact with the DOM (Document Object Model).
N	Non-functional Requirements: Additional features of the system provided to increase the user attraction
O	Object: A concept, abstraction, or thing with sharp boundaries and meaning for an application
S	System: A group of independent but interrelated elements comprising an integrated whole.
T	Technology: Any mechanical or electrical means to automate or improve manual work.
U	Use Case: Representation of the business processes of the system.
W	Web Based: An application that is accessed via a web browser over a network such as the Internet or an intranet.
X	XML : EXtensible Markup Language was designed to carry data, not to display data