HOTEL MANAGEMENT SYSTEM FOR GAYANA HOTEL AND BEACH RESTAURANT

Y.N.NIRMANI

2017

HOTEL MANAGEMENT SYSTEM FOR GAYANA HOTEL AND BEACH RESTAURANT

Y.N.Nirmani

BIT Registration Number : R110609

Index Number : 1106092

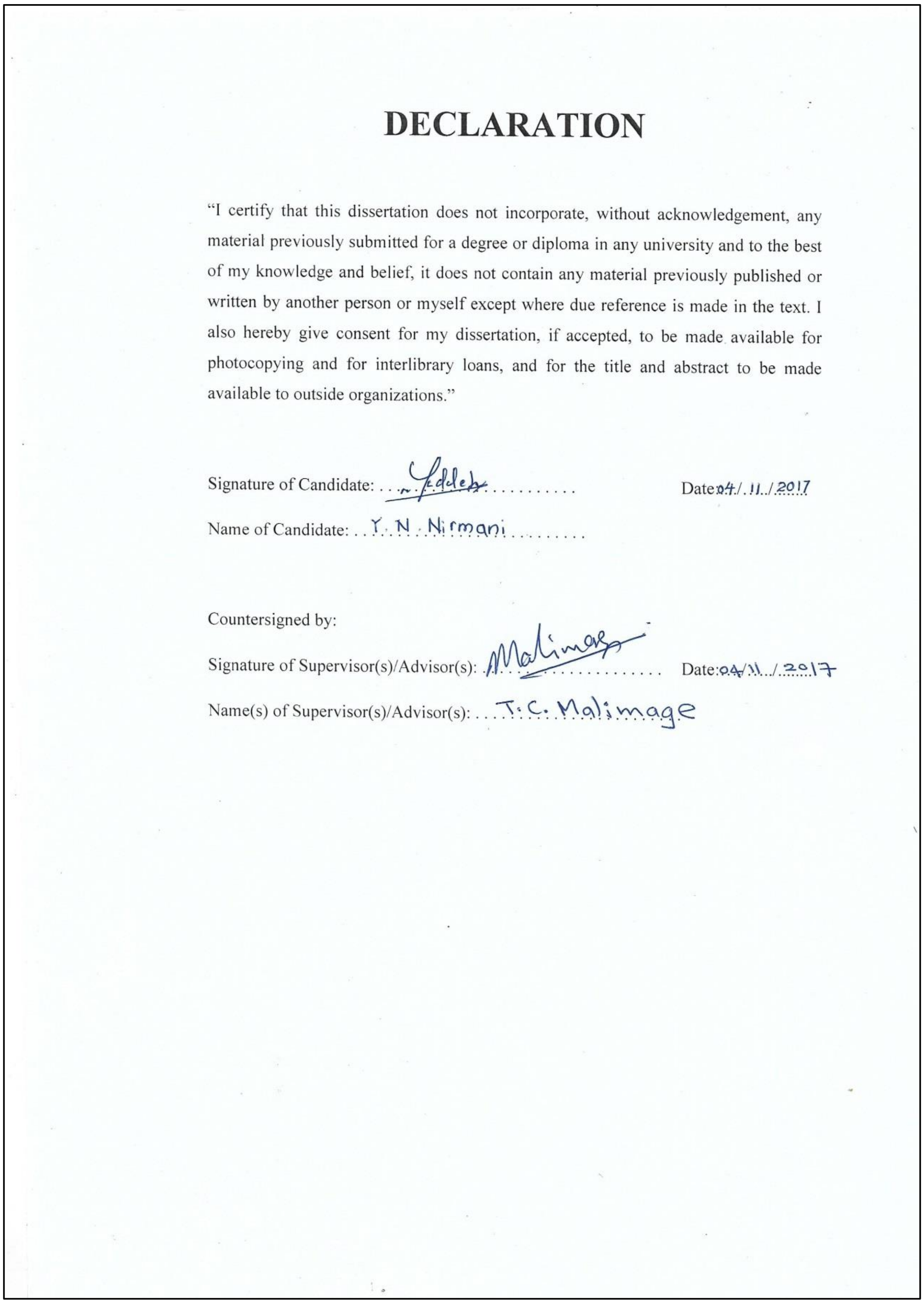
Supervisor: Mr.T.C.Malimage

2017

BIT

UCSC

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: [illegible] Date: 04/11/2017

Name of Candidate: Y. N. Nirmani

Countersigned by:

Signature of Supervisor(s)/Advisor(s): [illegible] Date: 04/11/2017

Name(s) of Supervisor(s)/Advisor(s): T. C. Malimage

ABSTRACT

Gayana Hotel and Beach restaurant is a small sized hotel which was started in 1990; it is one of the popular hotel in Hambantota district, which plays an important role in Sri Lankan tourism industry. As a hotel which interact with the customers daily it should have a systematic way to handle information and activities efficiently. Currently it has manual and paper based system to handle their day today activities. Due to this, hotel faces lot of inconvenience. As a solution hotel can get an aid from the information technology. The Hotel Management System is built to reduce and overcome these problems and will increase the competency and efficiency of their work.

The proposed system is develop to cover the process of the hotel such as Human resource management, Resource management, Services management, Package management, Notification management, Analysis management and User account management.

The system has adapted Model View Controller (MVC) architecture and object oriented techniques. In addition Unified Modeling Language (UML) was used for analysis and design. Hypertext Pre Processor (PHP) which is a server side scripting language was used to build the system. Also it was selected XAMPP as the development environment for the system, MySQL as the database environment and apache servers as the web server. The coding of the system was carried out using the NetBeans IDE 8.0.1. Specific design principles and technologies were used to develop the system.

The System has been developed to fulfill the user and system requirements deal with new technologies and enable hotel to face competition in the industry. It could also help the hotel administration to work efficiently and enable profit maximization through this newly introduced web based system.

AKNOWLEDGEMENT

First and foremost respectively, I greatly indebted the BIT Coordinator and academic staff of University of Colombo, School of Computing for giving the opportunity to undergo through this project and for knowledge gained.

I would like to extend my grateful acknowledgement to my supervisor Mr.T.C.Malimage for the given encouragement, gentle guidance and invaluable support offered through out of this study.

It is my foremost duty to pay my gratitude to my client, Management of Gayana Hotel & Beach Restaurant for the endorsement they gave me during the research period. Also I would like to thank staff members of Gayana Hotel & Beach Restaurant to been supportive to me from requirement analysis stage to testing stage. I express my indebtedness to all participants in the survey and all the participants of interviews for their encouragement, contribution, values and ideas, valuable assistance and especially their commitment towards this project and the valuable time they spent to make this project successful.

Finally, I am most grateful to parents and family for the never-ending support, encouragement throughout the study period and for being with me and make me relax at difficult times during this piece of work.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLE

LIST OF ACRONYS

1NF	-	First Normal Form
2NF	-	Second Normal Form
3NF	-	Third Normal Form
BIT	-	Bachelor of Information Technology
CD	-	Compact Disk
CSS	-	Cascading Style Sheet
HMS	-	Hotel Management System
HR	-	Human Resource
HTML	-	Hyper Text Markup Language
JS	-	Javascript
MVC	-	Model View Controller
NetBeans IDE	-	NetBeans integrated development environment
PHP	-	Personal Home Page / Hypertext Preprocessor.
SMS	-	Short Message Service
SQL	-	Structured Query Language
URL	-	Uniform Resource Locator
UML	-	Unified Modeling Language
XAMPP	-	Operating system,Apache, MySQL, PHP, and PERL

CHAPTER 1 - INTRODUCTION

1.1 INTRODUCTION TO THE GAYANA HOTEL & BEACH RESTAURANT

Gayana hotel & beach restaurant is located in a tourist area in Hambantota district. It is popular for among local and international tourists and wedding couples for banquet halls. It has been inaugurated in year 1990, through the past 27 years the hotel has been established as one of the best beach hotels in Hambantota area.

1.2 MORTIVATION FOR THE PROJECT

Gayana hotel & beach restaurant handles their all business operations such as keep records regarding the customers, employees, bookings, services, resources, packages and more other basic housekeeping functions manually by group of employees. Room handling, reception hall handling, services handling make conflicts and waste time. Both local and foreign guests reserve rooms and banquet halls, get services of it. Hotel has to keep all these records in a file based system. Keeping files takes much time.

Hotel has to manage number of employees with different job titles and salary scales. To approve employee leaves managers have to check previous leaves and sometime these details not up to date. Handling them manually caused lot of problems.

Hotel handles their bill calculations manually. So can't trust the accuracy of calculations and take much time. Customers face lot of problems when searching for bookings. If management wants to check for a previous room record or reservation details it will consume more time and inefficient Generating reports manually is more time consuming. Also it is difficult to keep standard of reports.

Hotel faces lots of difficulties during their operations because handling all the activities manually with a large number of users, resources and services create errors, conflicts, difficulties and time consuming. Hotel identified manual housekeeping activities are not reliable, accurate and inefficient and essential to have a computerized Hotel Management System.

Therefore an automated Hotel Management System for Gayana hotel & beach restaurant is proposed. So this system will be developed to support hotel to overcome above mentioned difficulties and do their operations successfully.

1.3 OBJECTIVES

The aim of this project is to create a Hotel Management System for Gayana hotel & beach restaurant.

The main objectives of this attempt are listed below;

- Develop a system that can replace the manual hotel management system.
- Improve efficiency and effectiveness of the Hotel.
- To record every transactions in computerized system.
- To design a user friendly graphical interface which suite the users.
- Improve the user interaction.
- Develop separate user loggings and control access.
- Give reliable search facility for the users.
- To save cost and time of the users.
- Provide to store all details regarding hotel.
- To eliminate the paper work.
- Implement cohesiveness and decoupling between functionalities
- Propose online working environment for the employees

1.4 SCOPE

The proposed system will assist the users in conducting his/her daily activities, as mentioned bellow. These activities will implement the system.

The scope for this project includes,

- Human resource management module
    - Manage employee detail (Add /Edit/Update)
    - Leave management (apply leaves, approve leaves, notify leave validity)
    - Manage system accounts
    - Handle employee account
- Resource module
    - Add/edit rooms
    - Add/edit room types
    - Add/edit reception halls

- Service module
    - Add/edit service
    - Add/edit service type
    - Add/edit service resource
- Package module
    - Add/edit customer
    - Handle bookings
    - Calculate total bill /payments
- Notification module
    - Make reminders and notifications
- Analysis module
    - Generate monthly reports
    - Make decisions for future
    - Check for profit in each month
    - Check for profits in each service
- User module
    - Add/edit user accounts
    - Assign privileges

1.5 STRUCTURE OF DISSERTATION

The dissertation contains six main chapters to convey the efforts that went into creating the Hotel Management System for Gayana hotel & beach restaurant. Each chapter will consist of required details in order to understand the project with the help of appropriate figures, graphs and reports. Dissertation structure is as follows;

- CHAPTER 02: ANALYSIS

The analysis chapter explains about the fact gathering techniques and collected functional, non-functional requirements to design and develop the system in detail. The existing system is presented here and an analysis of the problem is presented along with detail requirements of the system. Use case diagrams are used for a better understanding of the system as a system designing tool.

- CHAPTER 03: DESIGN

The information acquired in the analysis phrase is used to design a system that will satisfy the requirements of the client during the design phrase. In this chapter described the design methodologies of the system, tools and techniques used in the designing phase and the database design and user interfaces in detail.

- CHAPTER 04: IMPLEMENTATION

In implementation chapter described about the development process, implementation environment with the coding of the modules, the creation of the databases and user interfaces. Further, software tools and technologies used at the time of development and platform dependences are explained.

- CHAPTER 05: TESTING AND EVALUATION

The evaluation chapter described how the system was tested using sample data and the outcomes will be discussed to verify and validate the designed system. In here unit test and integration tests carryout and described the effects of various kind of errors and how the system was modified will also be reported here as well.

- CHAPTER 06: CONCLUSION

This is the final chapter of the dissertation, and includes the critical evaluation of the system and suggestions for any future work.

CHAPTER 2 - ANALYSIS

An analysis of the requirements provide in this chapter along with functional and non-functional requirements and fact gathering techniques.

2.1 FACT GATHERING TECHNIQUES

Requirement analysis was the most important process for clear identification of existing system and functional and non-functional requirements of the system. Various techniques were employed here to capture requirements from different user perspectives. The main methods used for fact finding process were;

- Interview
- Observation
- Document review

The most important factor involved in interviewing is to select the suitable people to be interviewed. The people selected to be interviewed should have good experience of the current manual system. Therefore Managers, selected employees, selected customers were interviewed as they were one of the primary users of the current manual Hotel Management System. It would be a big help to understand each process in detail and helped to identify the real situation of each process and real responsibility of each role. So this it could find new problematic areas of total process. By carrying out interviews gathered new requests related to the existing problems.

Throughout direct observations done to get an idea about the current manual system and identify administrative functions in the hotel. By observations identify the sequence of functionalities, interaction between main processors, main input and output of each process and also collected relevant information which will be used for the new suggested system. Some Hotel Management Systems were observed to identify functions of computer based Hotel Management System as a prototyping model.

Relevant documents such as Reports, journals and other formal paperwork documents of the existing manual system were selected, analyzed and studied through a sampling process to gather client requirements.

2.2 ANALYSIS THE CURRENT MANUAL SYSTEM

The requirements were analyzed through fact gathering techniques. System behavior is documented in a use case model. Use cases provide major inputs when finding and specifying subsystems, interfaces and test cases. The hotel management system has been divided into several logical subsystems in order to get a clear picture of the main processes.

2.2.1 REGISTRATION PROCESS

The current manual system uses paperwork and direct human communication to manage the hotel. The registration process of new employees to the hotel carried out by the HR manager by filling a form including personal details. Booking is done through phone calls or through visit to the hotel. When customer made a booking customer details entered to a registration registry by receptionist. After that confirm the booking by making payments and issuing recite.

2.2.2 HANDLING PAYMENTS

Payments will calculate according to the services booked by the customer. If the customer make a reception hall booking full payments should do before the booking date. If the customer makes a room booking can handle it before leaving.

2.2.3 SENDING REMINDERS

In manual system it’s not efficient and clear. Only through phone call remind the date to the customer.

2.2.4 INSERT/DELETE/UPDATE PROCESS

Hotel manually handles all the insertions, updates and deletions. Receptionist enters all records in registration books and others. When completed a booking that files remove from the front office.

2.2.5 REPORT GENERATION

The hotel manager should prepare reports on several criteria. So every time reports generate according to the need of the hotel. When doing it manual time consuming and inefficient.

2.3 FUNCTIONAL REQUIREMENTS

The functional requirements are the system functionalities that the Hotel Management System should achieve. Here are the functional requirements for the proposed system as identified through analysis. [1]

- User intendant search will be efficient and without time wasting.
- Salary calculation can be done accurately and effectively.
- Payment calculation and handling will be more accurate.
- Leave application and leave approval will be fast, accurate and efficient.
- Employees and customer data can be stored and updated accurately and effectively.
- Customers can get information quickly.
- Allow the administrator or super user to create user account and assign different privileges for different users.
- Managers use computer generated reports to future planning.
- Hotel details (resources, services, prices, location etc..) available to internet users 24 hours and 365 days.
- Reminders should be sent using email/SMS.

2.4 NON-FUNCTIONAL REQUIREMENTS

Not only the functional requirements but also the non-functional requirements are vital to successful implementation of a project. Non-functional requirements specify the important properties and constraints that the system should keep, in order to provide a best performance. In addition the system unable to meet non-functional requirements it may cause the whole system unusable.

- The system should be accurate and consistent. When manipulating the fed data in proper way and displaying correct information, the records should be consistent and not isolated. In this project registering employees/customers, updating details, calculating payments should be accurate and consistent.
- The system should be reliable. Only authorized persons should be able to give permission to log in to the system, view data and update data. Keeping backups is important in this process.
- The system should be secure. By limiting the access privileges only the authorized persons should be able to log in to the system, view data and update data. User loggings should create with access privileges.

- The system should be user friendly. Interfaces, links and menus should locate in proper places and navigation through the system should easy to the users. Non IT professionals can be able to use the system easily.
- The system should provide maintaining features of the system and ability to recover from system failures. In future alternate functions and requirement should be added to the system without major impacts on design.
- The system should be Portable. The system could be installed without effort and support major changes regardless of the machine location and operating system.

2.5 EXISTING SIMILAR SYSTEMS

By studying existing similar systems gained more experiences how to develop the system with required functionalities. Following are a few similar systems that were reviewed to build the system.[2]

1. Ezee PMS Hotel Solution System.

eZee FrontDesk can be classified as one of the most prominent hotel management software (HMS), Resort booking software, hotel accounting softwares, top Hotel PMS, resort software, property management software, Hospitality Management System, lodging software, hostel software and condo software.

Their reservation manager module is capable of doing checking availability, search reservation and front end operations. Also the system has separate modules for booking which handles the room reservations, channel manager and separate mobile applications. Also they provide user specified interfaces for their application.

The main drawback of this system is the complexity. User need high end computers to use this system and small and medium hotels and restaurants may find is more complex for their day to day operations. Also the cost is comparatively high.

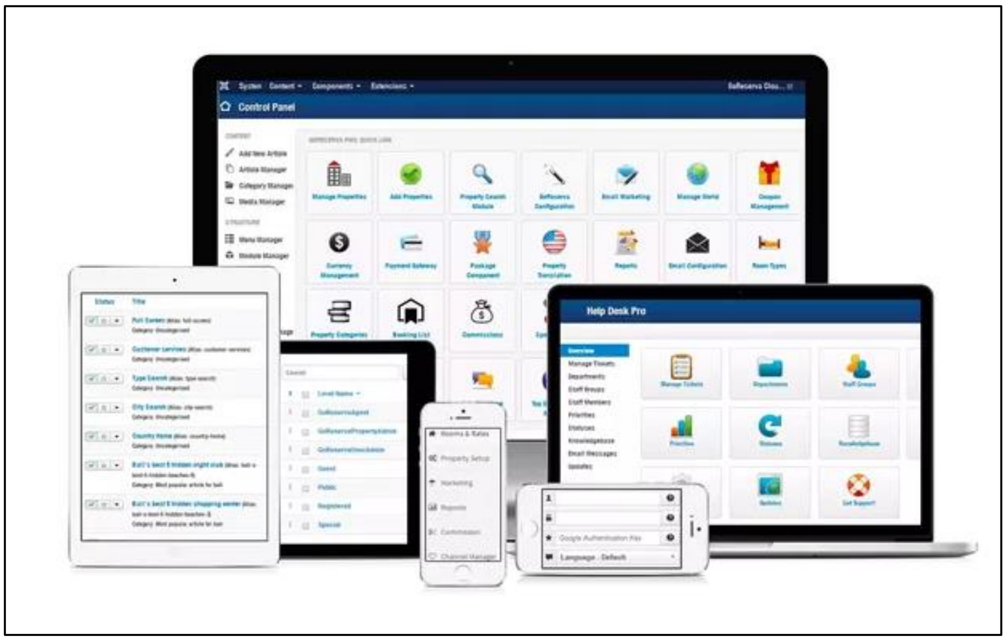


Figure 2.1 diagram of Ezee PMS Hotel Solution System

2. Hotel-line

Hotel-line is one of the leading software and it emerging very fast due to its ease of use and its user friendly interface. Hotel-line has a feature of multi property that provide easy view of all property from single login it present consolidated report with business comparison. Hotel-line software comes with new technology which becomes trends in the market. This hotel management system has channel management technology which not only handles online travel agent but is also maintain booking and make right amount of booking for the hotel. It communicates thousands of online ota, gds, booking system on the Internet for this purpose. Hotel-line has some well-defined and standard reports and in additional customized reporting tool where user can make unlimited custom report according to his requirement. Hotel-line hotel management system has wide range of module which full fill the requirement of property including restaurant, housekeeping department, bar, club, spa, banquet etc.

The software has been created by using latest responsive web design technology and assimilation of industry experts opinion aimed to enhance usability of the designed product. The major modules of our software include check-in/check-out, 2-click guest reservation, housekeeping, group management, travel agent, night audit, guest history and a whole lot of others for improving the efficiency and performance of all business operation aspects.

One of the major drawbacks of this system is the price and the high complexity for small and medium scale hotels and restaurants.

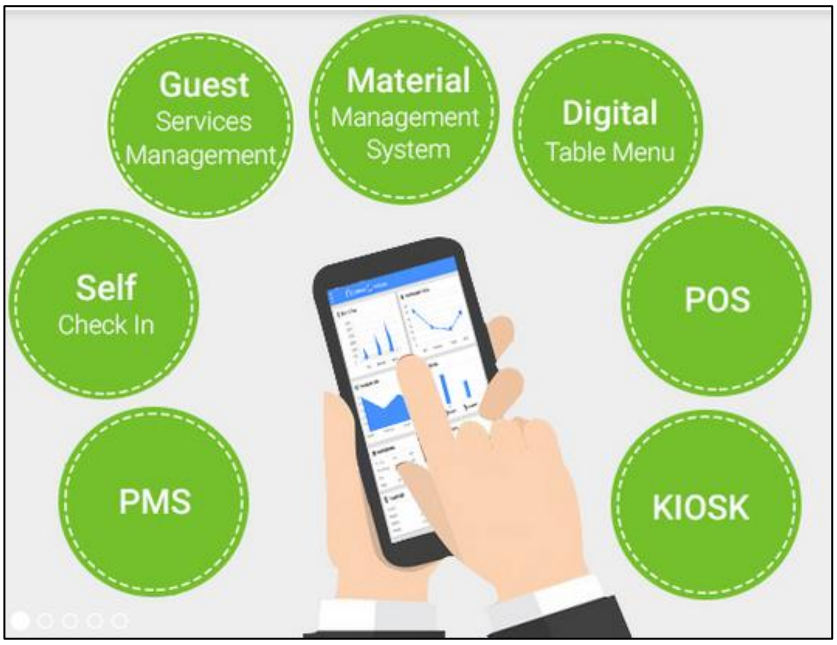


Figure 2.2 diagram of Hotel-line

CHAPTER 3 – DESIGN

Software designing phase is the most important phase in software development. Because the system is build up according to the design. So the system requirements which were gathered in analysis convert to a model of actual system. Here a “blue print” is made according to analyzed requirements. Software design consist of several steps, this includes both low-level component and algorithm design and high-level, architecture design.[3]

3.1 PROCESS MODEL

A process model is a modeling methodology followed throughout the development of the system. In this system mainly iterative processing model was followed. This was helped to develop the system in iteratively in small time period with clear understanding of the requirement correctly. And other few models used over the iterations like Waterfall model, Prototyping model and Rapid Application Development model. These models are briefly described below;

3.1.1 WATERFALL MODEL

The Waterfall Model was first Process Model to be introduced. It is also referred to as the linear-sequential life cycle model, suggests a systematic, sequential approach to software development that begins with the requirements of customer specified and progresses through planning, modeling, construction, and deployment also referred to as a. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discards the project.[4]

3.1.2 PROTOTYPING MODEL

The basic idea of prototyping is that a working model that is functionally equivalent to a component of the product before a design or coding can proceed. This prototype is developed based on the currently known requirements. By using this prototype, the client can get an actual feel of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired system. The developmental process only continues after the client is satisfied with the functioning of the prototype. At that stage the developer determines the specifications of the client’s real needs.

3.1.3 RAPID APPLICATION DEVELOPMENT MODEL

It is a type of incremental model that emphasizes a very short development cycle. In RAD model the components or functions are developed in parallel as if they were mini projects. The developments are time boxed, delivered and then assembled into a working prototype. It can quickly give the customer something to see and provide feedback regarding the delivery and their requirements.

3.2 METHODOLOGY FOR THE PROPOSED SYSTEM

Iterative enhancement model was used as the software methodology for the project. Since the project is customer oriented and project is medium scale, iterative enhancement fits well. Also it helps to get frequent customer feedback which helps to improve the quality and relatedness of the project.

3.2.1 ITERATIVE ENHANCEMENT MODEL

In iterative enhancement model, rather than deliver the system as a single delivery, the development and delivery is broken down into increments with each increment delivering part of the required functionality. User requirements are prioritized and the highest priority requirements are included in early increments. Once the development of an increment is started, the requirements are frozen though requirements for later increments can continue to evolve. Diagram of iterative enhancement model is depicted in figure 3.1 [5]

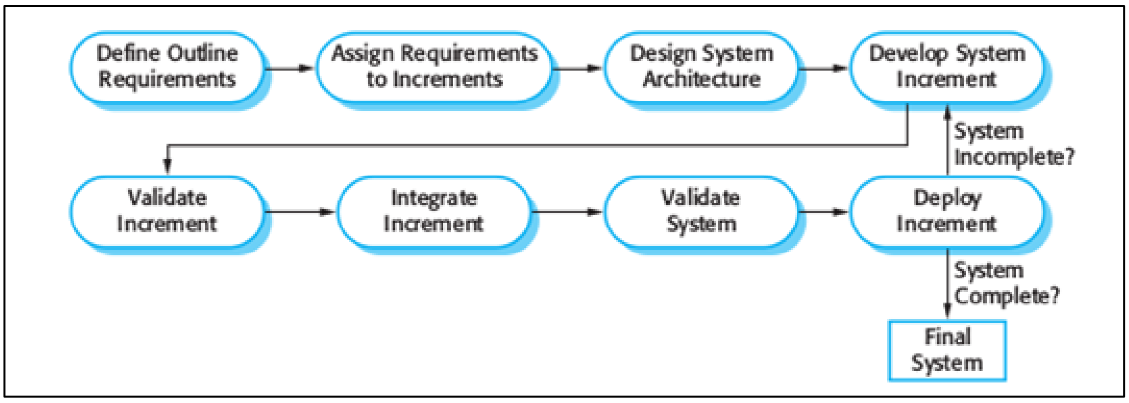


Figure 3.1 diagram of iterative enhancement model

There are high amount of advantages of iterative enhancement software model for a customer based web application. Customers can use the early increments as prototypes and gain experience that informs their requirements for later system increments. Unlike prototypes, these are part of the real system so there is no re-learning when the complete system is available. Customers do not have to wait until the entire system is delivered before they can gain value from it. The first increment satisfies their most critical requirements so they can use the software immediately. The process maintains the benefits of incremental development in that it should be relatively easy to incorporate changes into the system. As the highest-priority services are delivered first and increments then integrated, the most important system services receive the most testing. This means that customers are less likely to encounter software failures in the most important parts of the system.

For the project mostly used agile practices even though it only have a sole developer. The most widely adopted agile practices for this project are,

- Frequent customer feedback and delivery
- Test Driven Development
- Embrace Change
- Maintain Simplicity

3.3 ALTERNATIVE SOLUTIONS

The alternate solutions for the system are; developing a standalone system for the client, having a collection of software to perform these tasks or switch to the traditional manual file system.

3.3.1 STANDALONE SYSTEM

Standalone system is a system that does not require an operating system to run. Most standalone systems are platform dependent, so setting up the particular operating environment is very expensive. Since this is a standalone system only one person is required to operate the system at a time. It may cost when developing and time consuming.

3.3.2 TRADITIONAL SYSTEM

One of the alternative solutions is improvement of the manual system. The entire process based on the existing file based system can be used by the client. So it can manage all the work by using paper and generate reports and valuable feedbacks by analyzing those paper works when necessary. But this needs a lot of human resource and commitment. The hotel may need to hire more employees to do every work manually and it will not time effective.

3.4 REASONS TO CHOSE THE WEB BASED SYSTEM

The client particularly requested for a web based system, for monitor automate day to day activities.

- The client predominantly requested for a web based system.
- System is platform independent because users interact with the system via web browser.
- Easy deployment.
- Can be implemented on client-server architecture.
- Maintenance is easy, because the database is centralized and everything is synchronized.
- Easy to manage.
- Cost effective and timeliness.

3.5 PROPOSED ARCHITECTURE

MVC architectural pattern has existed for a long time in software engineering. MVC pattern stands for model, view, controller pattern. This pattern is used to separate applications concerns. Diagram of MVC architecture is depicted in figure 3.2 [6]

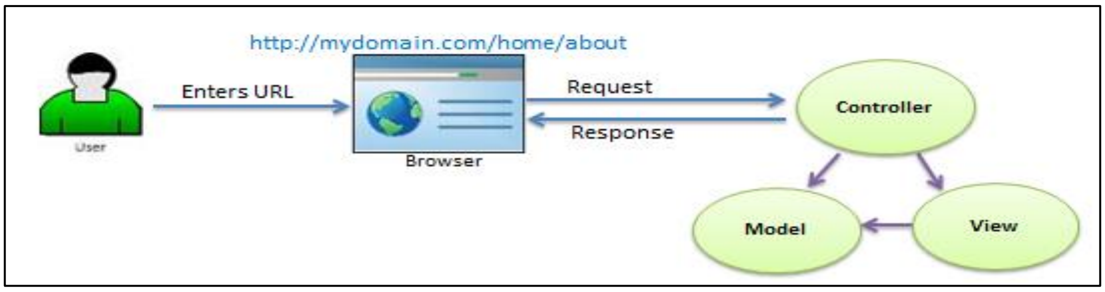


Figure 3.2 diagram of MVC architecture

Model:Model represents shape of the data and business logic. It maintains the data of the application. Model objects retrieve and store model state in a database.

View: View is a user interface. View display data using model to the user and also enables them to modify the data.

Controller: Controller handles the user request. Typically, user interact with View, which in-tern raises appropriate URL request, this request will be handled by a controller. The controller renders the appropriate view with the model data as a response.

3.6 OBJECT ORIENTED ANALYSIS AND DESIGNING FOR THE SYSTEM

The design process involves developing a conceptual view of the system, establishing system structure, identifying data streams and data stores, decomposing high level functions into sub functions, establishing relationships and interconnections among components, developing concrete data representations, and specifying algorithmic details. Software design is a creative activity. Design techniques define how the design process will be done and provide proper platform to represent the selected process methodology.

For the project, used UML (Unified Modeling Language).UML is a general-purpose, developmental, modeling language in the field of software engineering that will intend to provide a standard way to visualize the design of a system. It helps to represent the model in more abstract manner which helps to the understanding and decision making process. Following diagrams are used as tools in this process. [7]

- Use Case Diagram-use case diagrams overview the usage requirements for a system. They are useful for presentations to management and/or project stakeholders. Also it helps to understand the actions which the system performs at each sub system and by each actor.
- Class Diagram-Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, methods and the relationships among objects. The class diagram is the main building block of object-oriented modeling. It is used mainly for general conceptual modeling of the systematics of the application.
- Activity Diagram-Activity diagrams defines the dynamic behaviour of the system and helps the stakeholders to identify each action in more detailed manner rather than in abstraction.

- Sequence Diagram-Sequence diagram is a type of UML diagram which shows how objects operate with each other and specify in which order they interact. It shows object interactions arranged in a time sequence.

3.6.1 HIGH LEVEL USE CASE DIAGRAM FOR THE PROPOSED SYSTEM

Use case diagram is a most important diagram which represents the static view of the system. In addition it can be used to identify user requirements correctly. The high level use case diagram for the proposed system is depicted in figure 3.3. Please refer Appendix B for detailed information.

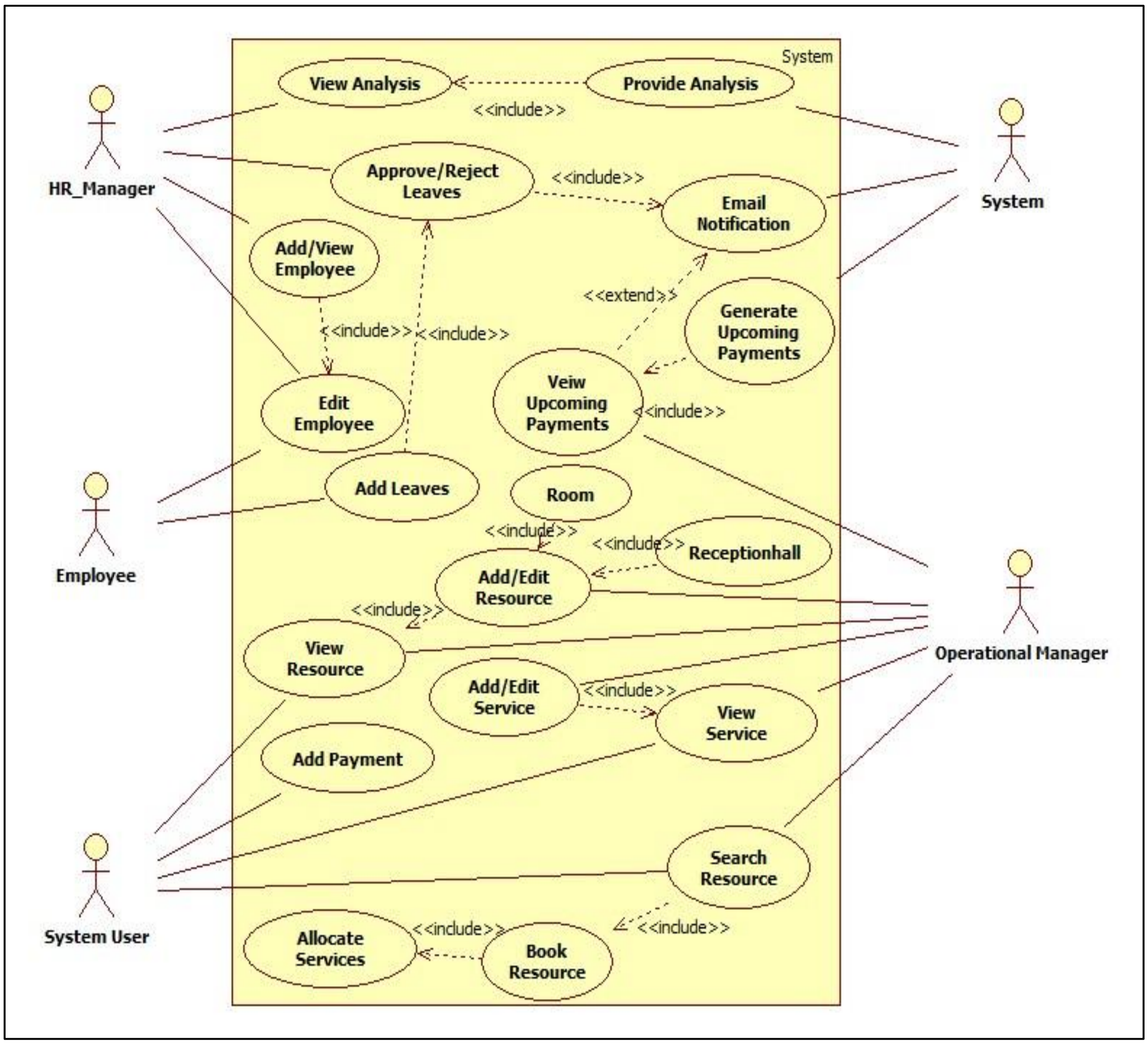


Figure 3.3 Use case diagrams for proposed system

3.6.2 CLASS DIAGRAM FOR PROPOSED SYSTEM

A class diagram shows a static or structural view of the system by showing the system’s classes, their attributes, operations and the relationships among objects. It does not show the dynamic nature of the communications between the objects of the classes in the diagram. Also Class diagrams are vital part of Object Oriented design method. They help to carry out conceptual / domain modeling. A conceptual model is a visual representation of conceptual classes or real-world in a domain of interest. Class diagram for proposed system is depicted in figure 3.4 [8]

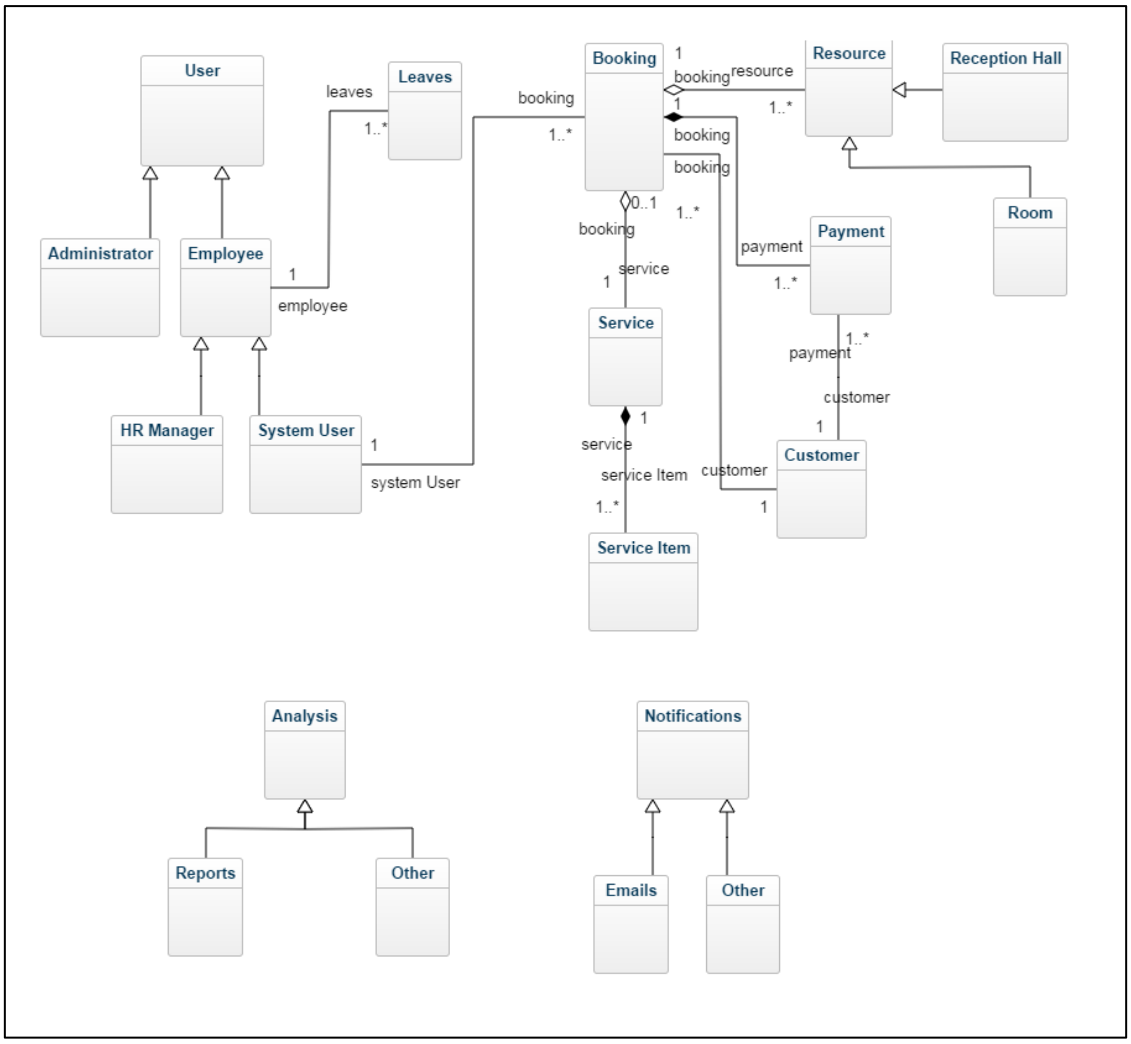


Figure 3.4 Class diagram for proposed system

3.6.3 ACTIVITY DIAGRAM FOR UPDATE EMPLOYEE DATA

A UML activity diagram depicts the dynamic behavior of a system or part of a system through the flow of control between actions that the system performs. Activity diagram for updating employee data is depicted in figure 3.5

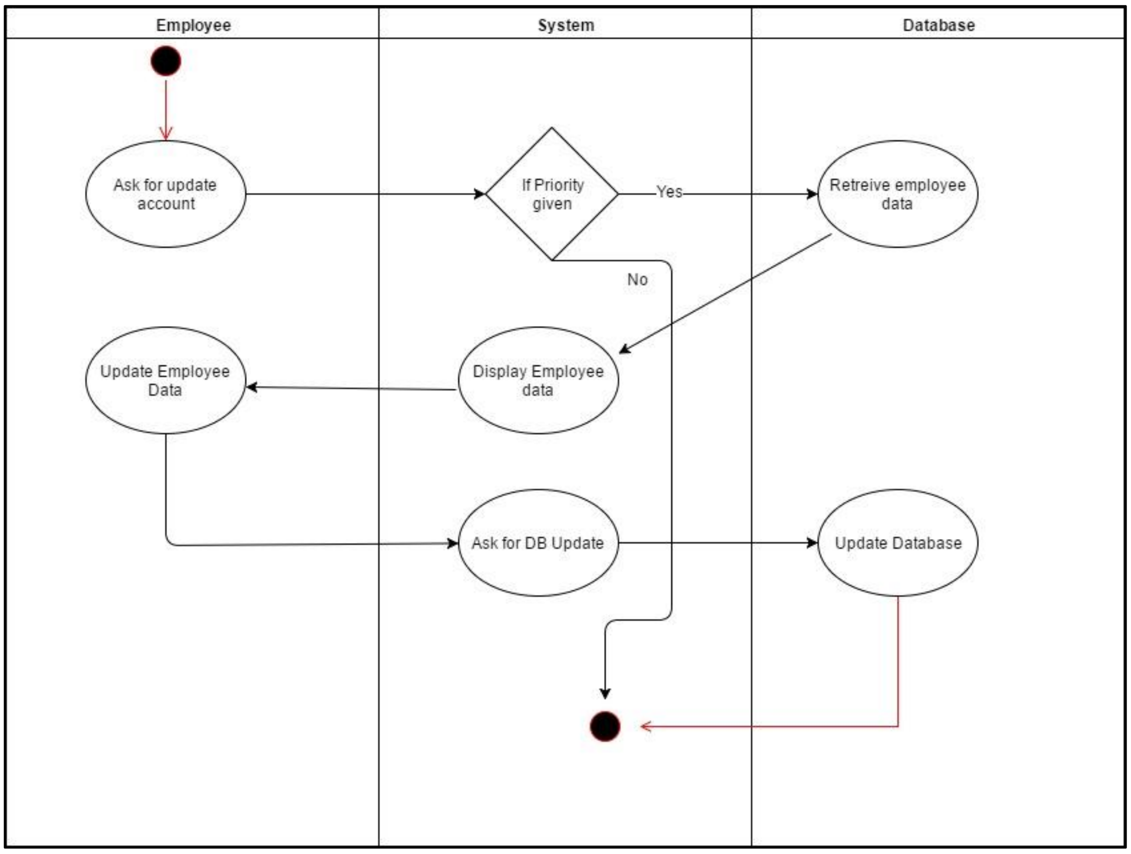


Figure 3.5 Activity diagram for update employee data

3.6.4 SEQUENCE DIAGRAM FOR LEAVE

A sequence diagram is used to show the dynamic communications between objects during execution of a task. Sequence diagram for leave is depicted in figure 3.6

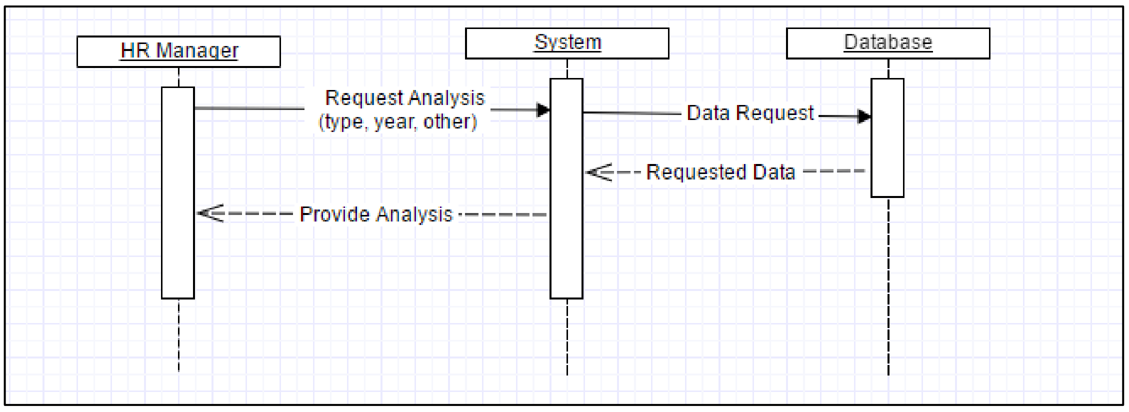


Figure 3.6 Sequence diagram for leave

3.6.5 SEQUENCE DIAGRAM FOR ADD EMPLOYEE

Sequence diagram for add employee is depicted in figure 3.7

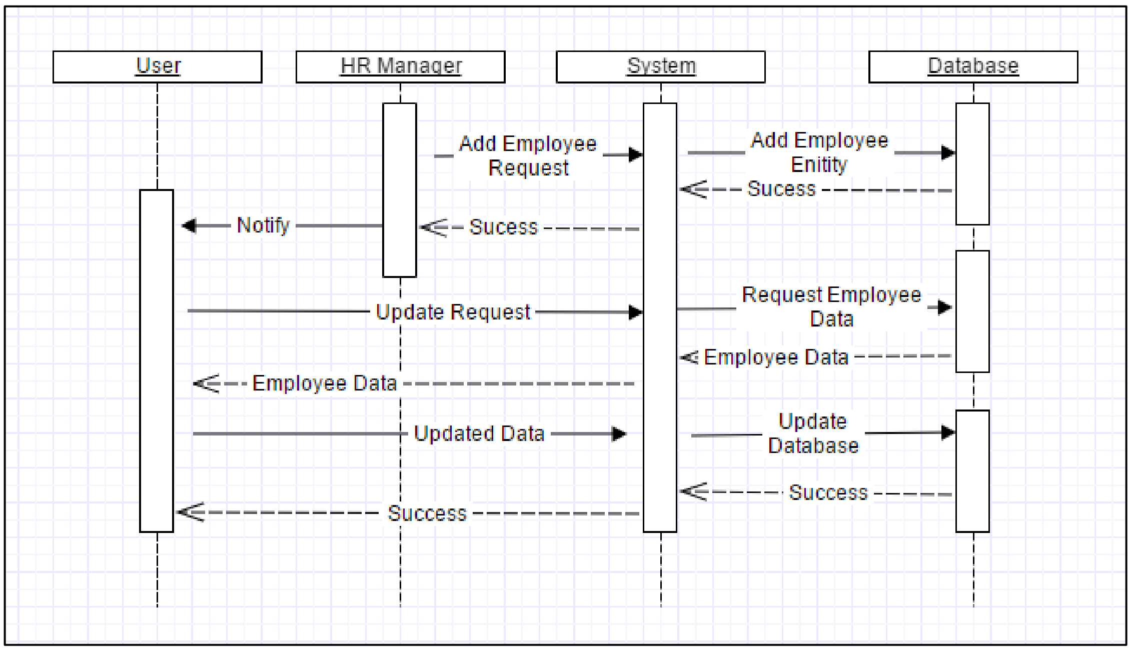


Figure 3.7 Sequence diagram for add employee

3.7 DATABASE DESIGN

This is a data driven system where providing easy access to volumes of data piled up on a long term basis is a key requirement of the system. This makes the database design a key aspect of the system. Database design is achieved through data modeling. This is a technique used for defining business requirements for a database. Database design is used to specify the structure of business objects used in the client server system. [9]

Database normalization is the process of organizing the fields and tables of a relational database to minimize redundancy and dependency. Normalization usually involves dividing large tables into smaller (and less redundant) tables and defining relationships between them.

- First Normal Form (1NF)

Main objective of the 1NF is eliminating the repeating groups and multi valued columns and arranges them in a single table, and defines a primary key for identifying each related attribute.

- Second Normal Form (2NF)

Main objective of the 2NF is eliminating the partial dependencies and creating separate tables and relate tables with a foreign key.

- Third Normal Form (3NF)

Main objective of the 3NF is eliminating the transitive dependencies.

Database diagram of the proposed system is depicted in figure 3.8

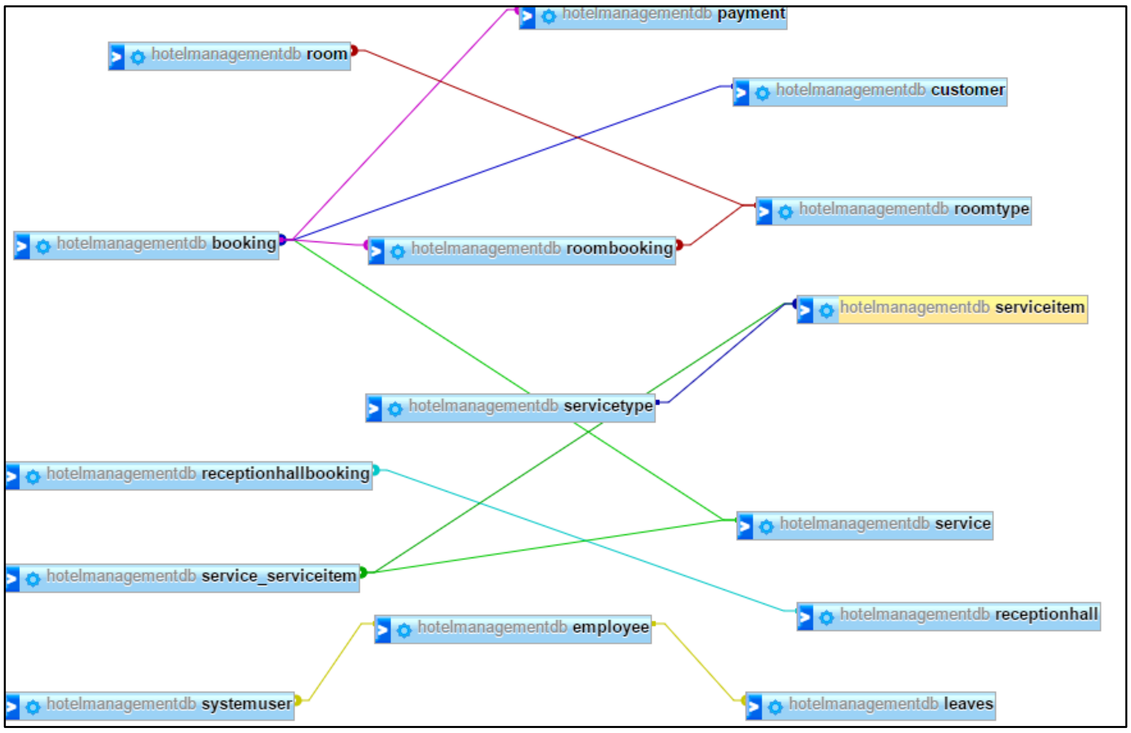

hotelmanagementdb payment
hotelmanagementdb room
hotelmanagementdb customer
hotelmanagementdb roomtype
hotelmanagementdb booking
hotelmanagementdb roombooking
hotelmanagementdb serviceitem
hotelmanagementdb servicetype
hotelmanagementdb receptionhallbooking
hotelmanagementdb service
hotelmanagementdb service_serviceitem
hotelmanagementdb receptionhall
hotelmanagementdb employee
hotelmanagementdb systemuser
hotelmanagementdb leaves

Figure 3.8 Database diagrams for the system

3.8 USER INTERFACE DESIGN

User interface is the main interaction between user and the system. Good interface design should match with user expectation. When designing user interfaces following properties were considered. [10]

- Simple
- Flexible
- Easy navigation
- User guidance
- Consistence

3.8.1 USER LOGIN INTERFACE

This is a common interface for all employee and system users. User have to select the user type before enter user details. At the beginning, user has to log in to system with employee id as the user name and password. Login interface enables user to reset password if it is forgotten. Different users have different privileges. Proper error handling and simplicity helps user to go ahead with the system. Login interface for the system is depicted in figure 3.9

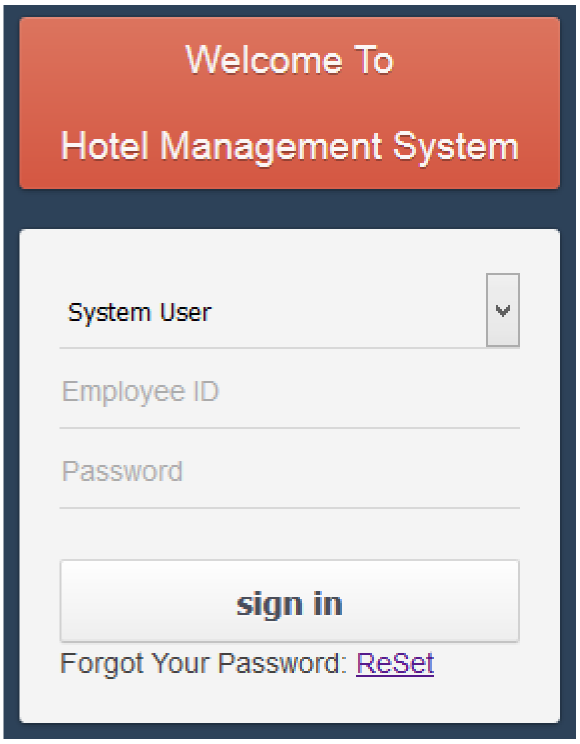


Figure 3.9 Login interface for the employee

3.8.2 MAIN MENU

Main menu changes according to the user type. Main menu for the system user login depicted in figure 3.10

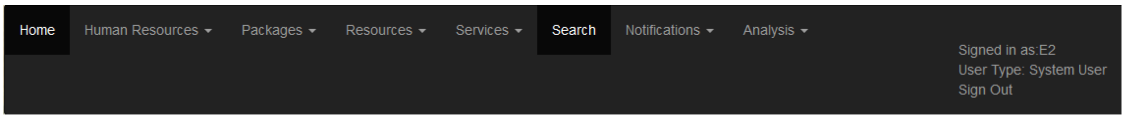
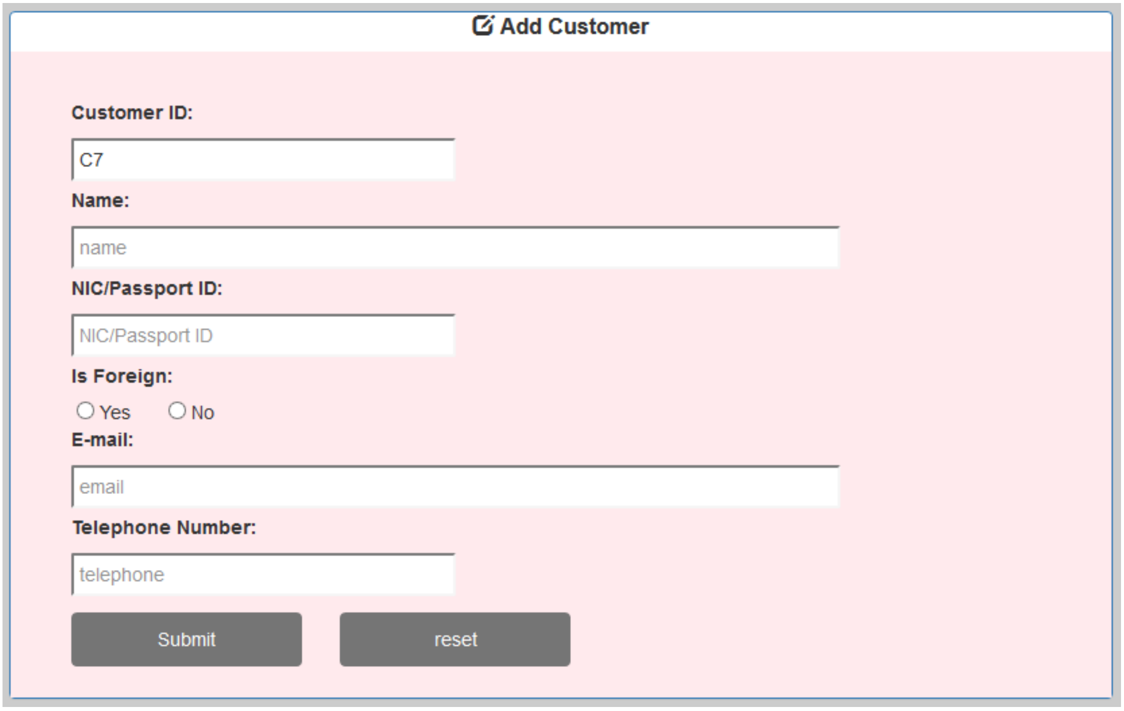


Figure 3.10 Main menu for the system user login

3.8.3 INTERFACE FOR ADD CUSTOMER

New customer can be entered through this interface. Before make a booking every customer should register in the system. Bookings can be done using customer ID. Add customer interface has depicted in figure 3.11

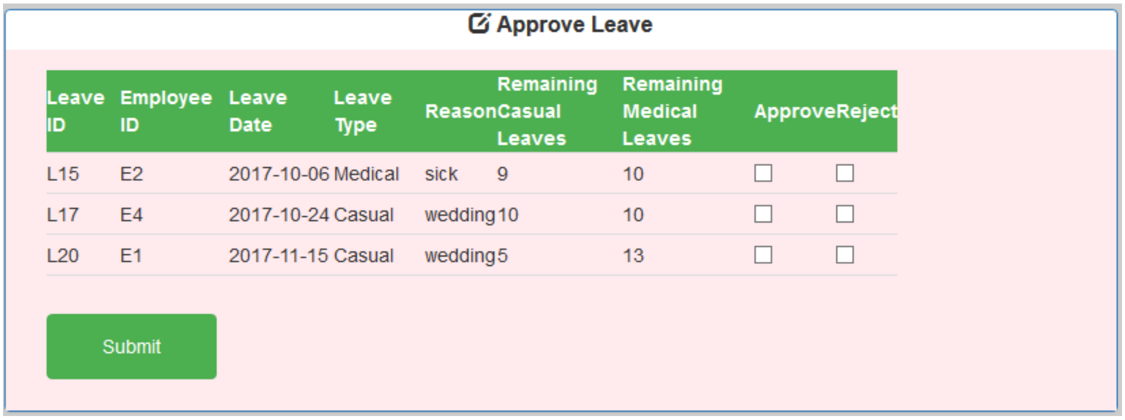

Add Customer

Customer ID:	C7
Name:	name
NIC/Passport ID:	NIC/Passport ID
Is Foreign:	○ Yes ○ No
E-mail:	email
Telephone Number:	telephone

Figure 3.11 Add customer interface

3.8.4 INTERFACE FOR APPROVE LEAVES

Leaves approval handle through this interface after employee make leave requests. Authorized system user can view, approve and reject leaves through this view. Approve leaves interface has depicted in figure 3.12

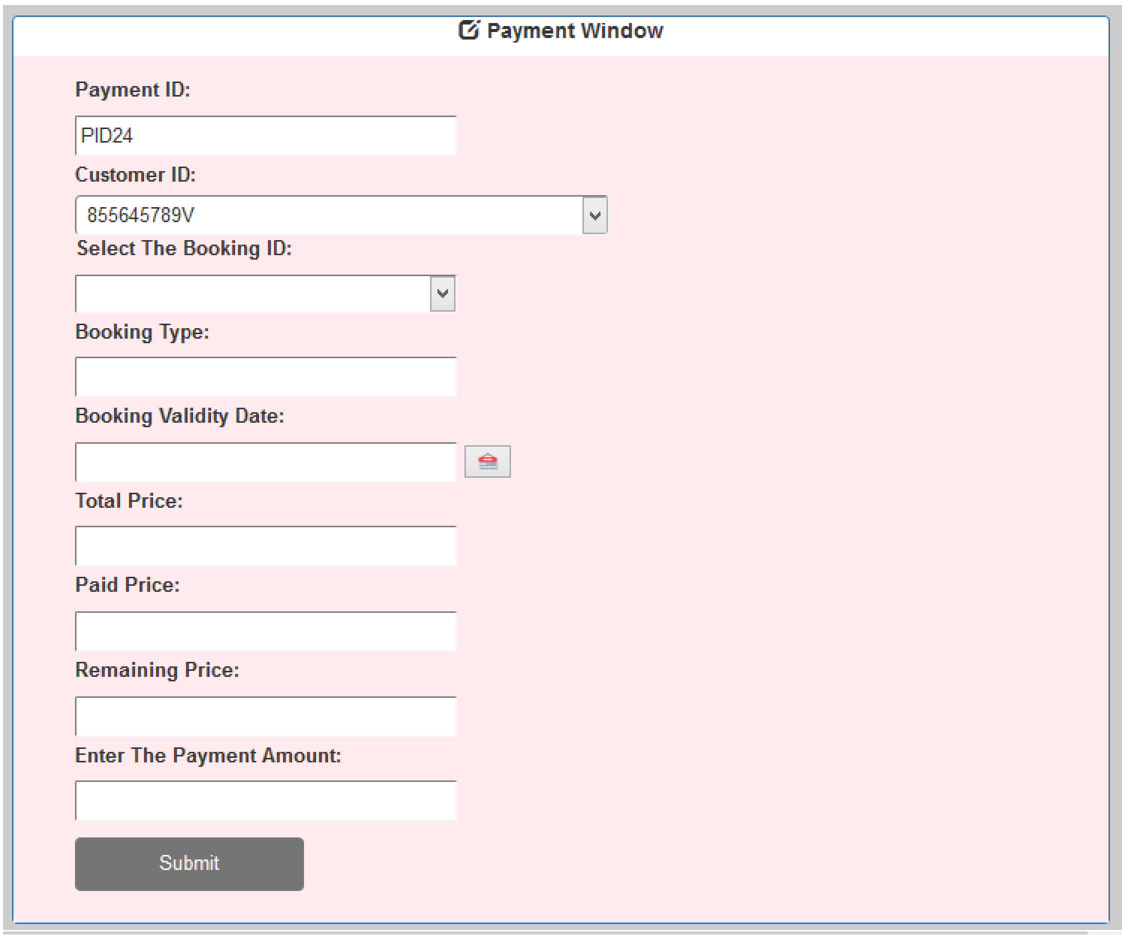

Approve Leave

Leave ID	Employee ID	Leave Date	Leave Type	Reason	Remaining Casual Leaves	Remaining Medical Leaves	Approve	Reject
L15	E2	2017-10-06	Medical	sick	9	10	☐	☐
L17	E4	2017-10-24	Casual	wedding	10	10	☐	☐
L20	E1	2017-11-15	Casual	wedding	5	13	☐	☐

Figure 3.12 Approve leaves interface

3.8.5 INTERFACE FOR HANDLE PAYMENTS

All payment details handle through payment window. Payment window interface has depicted in figure 3.13



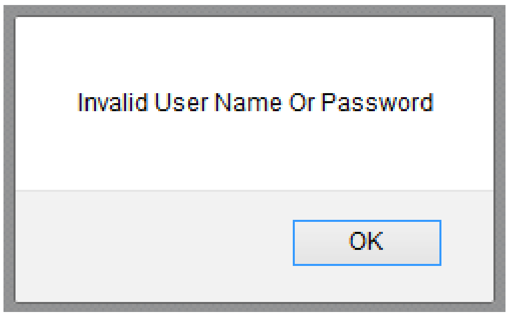
Payment Window

Payment ID:	PID24
Customer ID:	855645789V
Select The Booking ID:	
Booking Type:	
Booking Validity Date:	
Total Price:	
Paid Price:	
Remaining Price:	
Enter The Payment Amount:	

Figure 3.13 Payment window interface

3.8.5 MESSAGES

After saving records by adding or editing every form should show the meaningful messages. Message interface has depicted in Figure 3.14, Figure 3.15 and Figure 3.16



Invalid User Name Or Password

Figure 3.14 Error message for invalid login

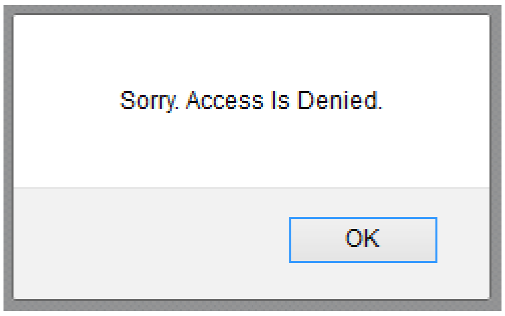

Sorry. Access Is Denied.

Figure 3.15 Error message for invalid privileges

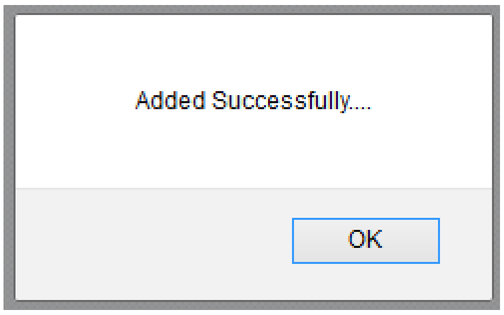
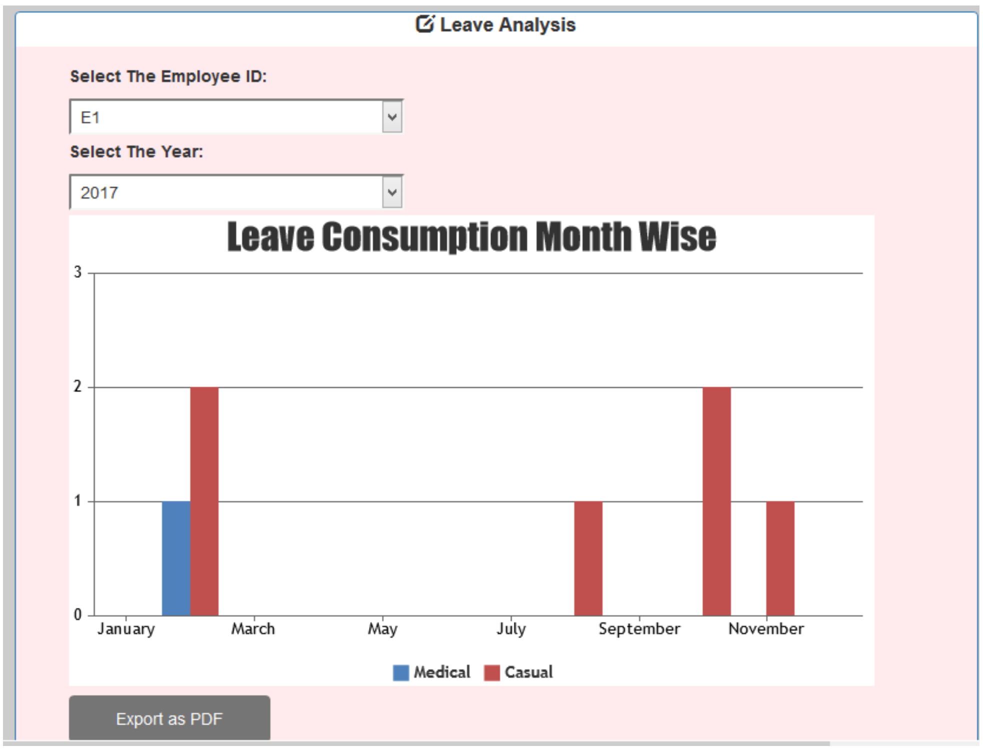

Added Successfully....

Figure 3.16 Successful message for adding details

3.8.6 REPORTS

Month wise analysis of employee leaves has depicted in Figure 3.17



Select The Employee ID: E1

Select The Year: 2017

Figure 3.17 Leave analysis interface

3.9 SECURITY OF THE SYSTEM

The privacy of the records is of extreme importance and a key constraint in designing the system. In order to maintain the required level of privacy and protection of data several mechanisms were introduced into the system.

- Password protection - Users must login to the system in order to access the system. A user name and a password are given to authorized users of the system.
- User access levels - The system has several user types and each type is given only required access, according to the existing methodology accepted by administrator. Some users can enter data but cannot delete it. Some users can enter, view and edit data. Some users have administrative access.
- Validation and verification - Validation and verification are implemented throughout the system to avoid loss of data due to human errors. For an example confirmation is required before deletion of a record and the system is checked before adding a new user to the system to prevent duplication of data. Validation is also carried out for many data fields.
- Data encryption - Authorizations needed for database access is stored separately in an encrypted format and used by the system after decrypting. This will prevent hacking into the system.

CHAPTER 4 – IMPLEMENTATION

Implementation is the process of converting the system specification in to an executable working system.

4.1 IMPLEMENTATION ENVIRONMENT

The system development environment can be classified as hardware environment and software environment. These two major categories listed below.

Hardware	Software
Intel Core 2 Duo Processor[2.40 GHz]	Apache
2GB RAM	MySQL
100GB Hard Disk	PHP 5
512MB Graphic Adapter	Tomcat
Internet connection	NetBeans IDE 8.0.1

Table 4.1: Implementation environment

The system was developed under windows 8 environment. But it is compatible with windows xp or newer version. [11]

4.2 SYSTEM DEVELOPED TOOLS AND TECHNOLOGIES

When developing the system following tools and technologies were used.

- NetBeans IDE 8.0.1
  NetBeans is open source platform that allows a software developer to create a customized development environment. The IDE also provides a great set of tools for PHP and C/C++ developers.[12]
- PHP [7.1.6 version]
  This is a server side object oriented scripting language which used when developing the system[13][14]
- MYSQL
  MySQL is an open source relational database management system used for handle database.[15]
- HTML
  This is the basic web related language and it helps to keep the system structure clear and conscious.[16]

- CSS

  The style sheet language is used to make styles for the webpages.[17]
- JavaScript

  Handle client side validation.[18]
- Ajax

  Support updating the system without refreshing.
- jQuery [19]

  This is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML

4.3 CODE AND MODULE STRUCTURE

The Hotel Management System was built on MVC framework which differs to programmer to programmer when it comes to implementation. Model view controller (MVC) is a software design pattern for implementing user interfaces on computers. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user. MVC interaction possible like in Figure 4.1

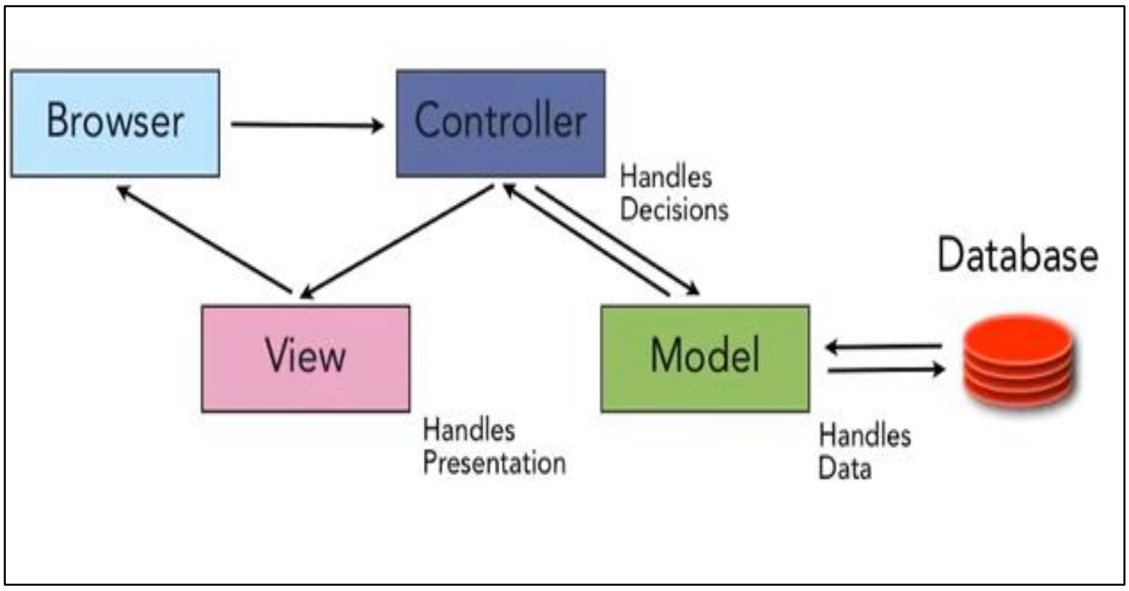
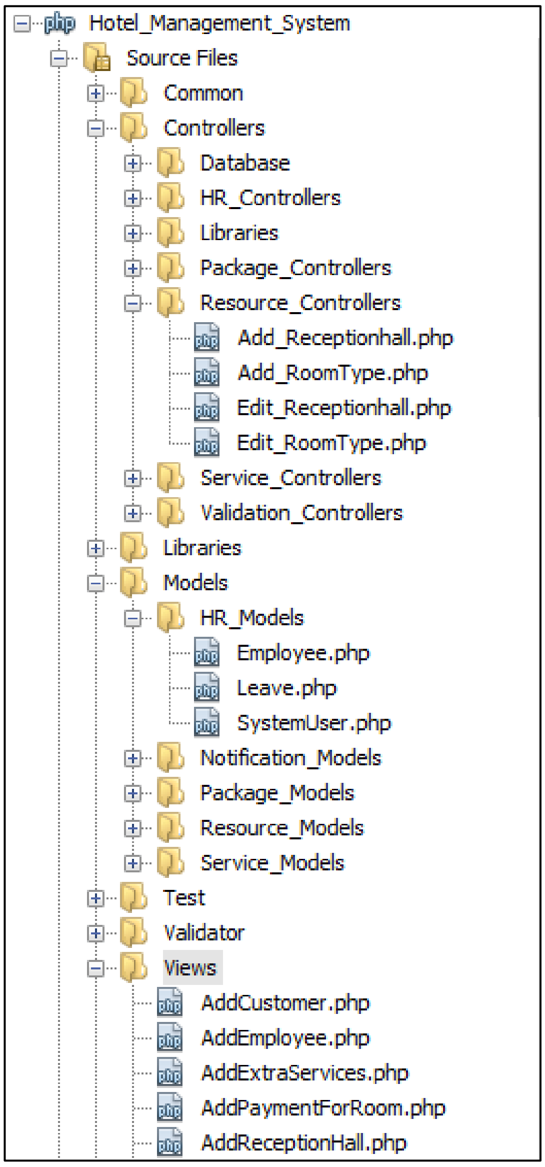


Figure 4.1 MVC sample interactive overview

The main code modules developed in the system have been mentioned below in figure 4.2.



```
Hotel_Management_System
    Source Files
        Common
        Controllers
            Database
            HR_Controllers
            Libraries
            Package_Controllers
            Resource_Controllers
                Add_Receptionhall.php
                Add_RoomType.php
                Edit_Receptionhall.php
                Edit_RoomType.php
            Service_Controllers
            Validation_Controllers
        Libraries
        Models
            HR_Models
                Employee.php
                Leave.php
                SystemUser.php
            Notification_Models
            Package_Models
            Resource_Models
            Service_Models
        Test
        Validator
        Views
            AddCustomer.php
            AddEmployee.php
            AddExtraServices.php
            AddPaymentForRoom.php
            AddReceptionHall.php
```

Figure 4.2 Structure of the code base

4.4 MAJOR CODE SEGMENTS

The main code modules of the system are mentioned here. There is brief description about the functionality of each code segment. Code modules consist of comments to identify the usage of important code lines.[20]

4.4.1 DATABASE CONNECTION PAGE

The system connects with a database. The data have to be inserted, updated, retrieved and deleted to and from the database. The given code show how the database connection has built up.

```php
<?php
class Connection
{
    private $_connection;
    private $_host = "localhost";
    private $_username = "root";
    private $_database = "hotelmanagementdb";
    public static function getInstance() {
    $conn
=mysqli_connect("localhost","root","","hotelmanagementdb");
    return $conn;
    }
    private function __construct() {
        $this->_connection = new mysqli($this->_host, $this-
>_username,
        $this->_password, $this->_database);
        if(mysqli_connect_error()) {trigger_error("Failed to
conencto to MySQL: " . mysql_connect_error(), E_USER_ERROR);
        }
    }
    private function __clone() { }
    public function getConnection() {
        return $this->_connection;
    }
                    function get_host() {
        return $this->_host;
    }
    function get_username() {
        return $this->_username;
    }
    function get_database() {
        return $this->_database;
    }
    function set_host($_host) {
        $this->_host = $_host;
    }
    function set_username($_username) {
        $this->_username = $_username;
    }
    function set_database($_database) {
        $this->_database = $_database;
    }
}
```

4.4.2 INDEX.PHP/LOGIN HANDING PAGE

This page is one of the main building blocks of the system. Where the base path is defined and the whole Hotel Management System URL is optimized.

```
<html>
      <head>
            <title>Registration Form</title>
            <meta charset="UTF-8">
            <meta name="viewport" content="width=device-
width, initial-scale=1.0">
      </head>
      <body>
  <h1 class="register-title">Welcome To The Login
Portal</h1>
  <link rel="stylesheet" href="..//Common/CSS/style.css" >
  <form class="register"
action="../Controllers/HR_Controllers/Signin.php"
method="post">
    <form action="<?php
htmlentities($_SERVER['PHP_SELF']);?>" method="post">
    <select name="type" id="type" class="register-input">
    <option value="1">System User</option>
    <option value="2">Employee</option>
     </select>
    <input name="username" id="username" class="register-
input" placeholder="Employee ID">
    <input type="password" name="password" id="password"
class="register-input" placeholder="Password">
    <button type="submit" class="register-button"
name="submit">Submit</button>
     <?php echo "Forgot Your Password: " ?><a
href="ReSet">ReSet</a>
  </form>
      </body>
</html>
```

Code segment given above gets the database connection and checks that the employee ID exists according to the input. If exists, then checks the password. If employee ID and entered password matched continue the process. Otherwise display an error message.

4.4.3 MODEL CLASSES

There are separate models to each identified entity in the system. They have necessary states (attributes) and behaviors (Getter and setter methods). Shown below is an example of System User class which simply includes the attributes and the Getters and Setters for those attributes.

```
class SystemUser {
    private $employeeId;
    private $userId;
    private $userPass
    private $priority;
    function __construct() {
    }
    function getUserId() {
        return $this->userId;
    }
    function getUserPassword() {
        return $this->userPassword;
    }
    function getPriority() {
        return $this->priority;
    }
    function setUserId($userId) {
        $this->userId = $userId;
    }
    function setUserPassword($userPassword) {
        $this->userPassword = $userPassword;
    }
    function setPriority($priority) {
        $this->priority = $priority;
    }
    function getEmployeeId() {
        return $this->employeeId;
    }
    function setEmployeeId($employeeId) {
        $this->employeeId = $employeeId;
    }
}
```

4.4.4 DATABASE CONTROLLER CLASSES

Database controller classes are used to connect with the database and exchange the data with the database. Object oriented approach is used for the implementation.

```
    public static function addCustomer(Customer $customer){
        $conn = Connection::getInstance();
        $customerid = $customer->getId();
        $name = $customer->getName();
        $nic=$customer->getNic();
        $isforeign = $customer->getIsForeign();
        $email=$customer->getEmail();
        $telephone=$customer->getTelephone();
        $sql="INSERT INTO
customer(id,nic,name,telephone,isforeign,email)". " VALUES ('"
. $customerid. "','" . $nic. "','" . $name. "','" .
$telephone. "','" . $isforeign. "','" . $email. "')";
    $result= mysqli_query($conn,$sql);
    return $result;
    }
```

4.4.5 CONTROLLER

Controller acts as a messenger between the models and views. It gets the requests and responses from each model and updates the view files according to the results. The code below show how Add employee function occurred.

```
if(isset($_POST["submit"])) {
        $username=$_POST["username"];
        $name = $_POST["name"];
        $employeeid = $_POST["employeeid"];
        $nic=$_POST["nic"];
        $birthday=$_POST["birthday"];
        $address=$_POST["address"];
        $usersname=$_POST["usersname"];
        $password=$_POST["password"];
        $email=$_POST["email"];
        $telephone=$_POST["telephone"];
        $gender=$_POST["gender"];
        $position=$_POST["position"];
        $recruitmentdate=$_POST["recruitmentdate"];
        $salary=$_POST["salary"];
        $loginuser=$_POST["username"];
        $logintype=$_POST["type"];
        $casualleaves=$_POST["casualleaves"];
        $medicalleaves=$_POST["medicalleaves"];
        $fileName = $_FILES["image"]["name"];
        $tmpName  = $_FILES["image"]["tmp_name"];
        $fileSize = $_FILES["image"]["size"];
        $fileType = $_FILES["image"]["type"];

        $fp      = fopen($tmpName, 'r');
        $image = fread($fp, filesize($tmpName));
        $image = addslashes($image);
        fclose($fp);
```

```
$employee=new Employee();
        $employee->setName($name);
        $employee->setEmployeeId($employeeid);
        $employee->setNic($nic);
        $employee->setUsername($usersname);
        $employee->setGender($gender);
        $employee->setPosition($position);
        $employee->setRecruitmentDate($recruitmentdate);
        $employee->setSalary($salary);
        $employee->setNoOfCasualLeaves($casualleaves);
        $employee->setNoOfMedicalLeaves($medicalleaves);
        $employee->setImage($image);
        $employee->setLoginUser($loginuser);
        $employee->setLoginType($logintype);

EmployeeDBController::addEmployee(Connection::getInstance(),$e
mployee);
}
```

4.4.6 VIEW

All the interfaces and presentation logics reside under views. The separation of logics in the views takes more effort. The variables play a reasonable role in the views because sometimes same forms are used for data insertion as well as the data updates. The given code show front end details of Add customer.

```
<html>
    <head>
       <title>Customer Registration Form</title>
       <script src="../Validator/CustomerValidator.js"></script>
       <meta charset="UTF-8">
       <meta name="viewport" content="width=device-width,
initial-scale=1.0">
     </head>
     <body>
     <h1>Add Customer</h1>
<formaction="..//Controllers/Package_Controllers/Add_Customer.ph
p" method="post" onsubmit="return validate()">
    <label>Customer ID:</label>
    <?php
    include ("..//Common/Connection.php");
    $conn = Connection::getInstance();
    $query ='SELECT * FROM customer' ;
    $result= mysqli_query($conn,$query);
    $num=  mysqli_num_rows($result);
    $num=$num+1;
    $id='C'.$num;
    ?>
    <input type=text" name="customerid" value=<?php echo $id ?>
readonly><br>
    <label>Name:</label>
    <input type=text" name="name" id="name" placeholder="name">
    <label id="name_err" > </label>
    <label>NIC/Passport ID:</label>
    <input type=text" name="nic" id="nic" br>
    <label>Is Foreign:</label>
    <label id="isforeign_err" > </label><br>
    <input type="radio" name="isforeign" id="isforeign_yes"
value="1"> Yes<br>
    <input type="radio" name="isforeign" id="isforeign_no"
value="0"> No<br>
    <label>E-mail:</label>
    <input type=text" name="email" id="email">
    <label id="email_err" > </label><br>
    <label>Telephone Number:</label>
    <input type=text" name="telephone" id="telephone">
    <label id="telephone_err" > </label><br>
    <button type="submit" name="submit">Submit</button>
    </form>
     </body>
 </html>
```

4.4.7 BUSINESS LOGIC FUNCTIONALITIES

Calculate Prices for Packages

This method provides specialized price calculations for weekdays and weekends for the rooms.

```
$query ="SELECT * FROM roomtype where type='$type' and
capacity='$capacity' " ;
    $result= mysqli_query($conn,$query);
     while ($row = mysqli_fetch_array($result)) {
          $normalPrice=$row['normalPrice'];
          $specialPrice=$row['specialPrice'];
     }
    $beginDate = new DateTime($checkin);
    $endDate = new DateTime($checkout);
    $price=0;
    $bookingdates=array();
    $daterange = new DatePeriod($beginDate, new
DateInterval('P1D'), $endDate);
    foreach($daterange as $date){
        $bookingdates[]=$date->format("Y-m-d");
    }
    $bookingdates[]=$checkout;
    for ($i = 0; $i < count($bookingdates); $i++) {
   $date1 = strtotime($bookingdates[$i]);
    $date1 = date("l", $date1);
    $date1 = strtolower($date1);
    if($date1 == "saturday" || $date1 == "sunday") {
      $price=$price+(float)$specialPrice;
    } else {
      $price=$price+(float)$normalPrice;
    }
    }
```

4.4.8 VALIDATION

Validate Date

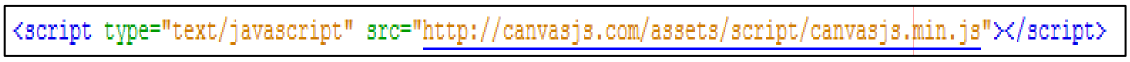
When requesting a leave by employee system check requesting leave date with the current date.

```
function checkDate(datestr) {
       var now = new Date();
       var dt1 = Date.parse(now);
       dt2 = Date.parse(datestr);
       if (dt2 < dt1) {
           return false;
       }
       else{
       return true;
   }
```

4.5 REUSABLE COMPONENTS

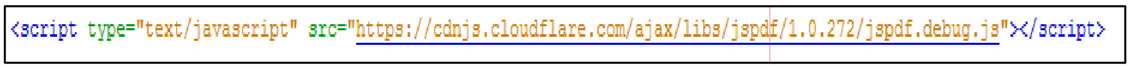
Re usable codes such as CSS, JQuery and others were used when building the system. Find them by referring internet. Those extracted codes were customized and well tested. Following are some of them. [21] [22]

- **Used canvasjs to draw charts.**



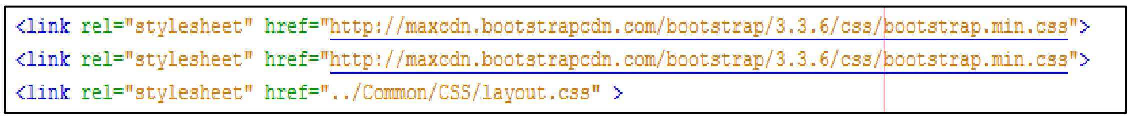
```
<script type="text/javascript" src="http://canvasjs.com/assets/script/canvasjs.min.js"></script>
```

- **Used jspdf to download files as pdf**



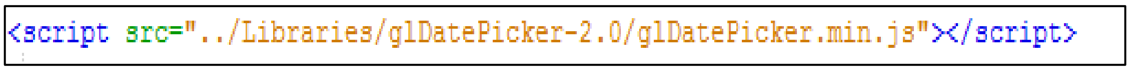
```
<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/1.0.272/jspdf.debug.js"></script>
```

- **Used CSS**



```
<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<link rel="stylesheet" href="../Common/CSS/layout.css" >
```

- **Used to set calendar**



```
<script src="../Libraries/glDatePicker-2.0/glDatePicker.min.js"></script>
```

- **email functionality**

These methods provide email functionality to send customers payment notifications, new offer notifications and other relevant information.

```
class EmailController {
    //put your code here
public static function sendEmail(string $email,string
$message,string $subject){
        $mail = new PHPMailer();
        $mail->IsSMTP();
        $mail->SMTPDebug = 0;
        $mail->SMTPAuth = true;   // Gmail requires
authentication
        $mail->SMTPSecure = 'ssl'; // Gmail requires SSL
connection
        $mail->Host = 'smtp.googlemail.com';
        $mail->Port = 465;
        $mail->Username = "projectnavoda@gmail.com"; // <--
Update this with your GMail username
        $mail->Password = "project123"; // <-- Update this with
your GMail password
        $mail->From="projectnavoda@gmail.com";
        $mail->FromName="Gayan Resorts Admin";
        $mail->Subject = $subject;
        $mail->Body = $message;
        $to=$email;
        $mail->AddAddress($to);
        $mail->Send();
    }
```

CHAPTER 5 – EVALUATION

Software testing is a process that should be done during the development process to verify the entire system accomplished its goals. The testing is comprised with verification and validation process. Verification refers whether the system implements the specified functions properly and satisfied the conditions. Validation refers to whether the system satisfied the requirements at the end of the development process.

5.1 TESTING TECHNIQUES

There are two types of techniques in software testing.

1. Black box testing

Black box testing is testing software based on output requirements and without any knowledge of the internal structure or coding in the programme.

2. White box testing

White box testing is testing software with the knowledge of the internal structure and coding inside the programme.

5.2 TEST PLAN AND TEST CASES

A test case is a set of conditions which a tester will determine whether a system under test satisfies requirements or works properly. The process of developing test cases can also help find problems in the requirements or design of an application. Test cases were planned and carried out for each module of the system which includes the test description, steps to test and the expected result. Once the errors found here were corrected. Some test cases designed for the important modules are as follows.

5.2.1 TEST CASES FOR LOGIN AUTHONTICATION

The test cases for the user login authentication module are given in table 5.1

Test number	**Test description**	**Steps to test**	**Expected results**
1	System User/ Employee log in to the system	Enter correct employee ID Enter correct password Click on sign-in	Successfully login to the system
2		Enter correct employee ID Enter incorrect password Click on sign-in	Display pop up message Invalid User name or password
3		Enter incorrect employee ID Enter correct password	Display pop up message Invalid

		Click on sign-in	User name or password
4		Enter incorrect employee ID Enter incorrect password Click on sign-in	Display pop up message Invalid User name or password
5	System User/ Employee forgot password and request to reset	Click on reset password Enter email Address submit	Receive a code through email to reset password

Table 5.1: User authentication test cases

5.2.2 TEST CASES FOR ADD EMPLOYEE

The test cases for the Add employee are given in table 5.2

Test number	Test description	Steps to test	Expected results
1	Add validate employee details	Add relevant details for each field in the form	Add the details to the employee table as a new row
2		Try to add details with empty fields	Display error messages relevant to each field.
3		Enter incorrect name	Display error message Invalid Name
4		Enter incorrect NIC	Display error message NIC
5		Enter incorrect email	Display error message Invalid Email
6		Enter incorrect telephone number	Display error message Invalid Telephone number
7		Enter incorrect salary	Display error message Salary should be numeric
8		Enter different passwords	Display error message Password do not match
9		Did not select birth day	Display error message Invalid birth day
10		Did not select recruitment day	Display error message Invalid recruitment day
11		Did not select gender	Display error message select gender
12	Submit employee details	Click on submit button to submit added details	Display pop up message Added Successfully

Table 5.2: Add employee test cases

5.2.3 TEST CASES FOR ADD LEAVES

The test cases for the Add leaves are given in table 5.3

Test number	Test description	Steps to test	Expected results
1	Add validate leave details	Add relevant details for each field in the form	Add the details to the leaves table as a new row
2		Try to add details with empty fields	Display error messages relevant to each field.
3	Submit leave request	Click on submit button to submit added details	Display pop up message request send for approval

Table 5.3: Add leaves test cases

5.2.4 TEST CASES FOR ADD CUSTOMER

The test cases for the Add customer are given in table 5.4

Test number	Test description	Steps to test	Expected results
1	Add validate customer details	Add relevant details for each field in the form	Add the details to the customer table as a new row
2		Try to add details with empty fields	Display error messages relevant to each field.
3		Enter incorrect name	Display error message Invalid Name
4		Enter incorrect email	Display error message Invalid Email
5		Enter incorrect telephone number	Display error message Invalid Telephone number
6	Submit customer details	Click on submit button to submit added details	Display pop up message Added Successfully

Table 5.4: Add customer test cases

5.2.5 TEST CASES FOR SEARCH ROOM BOOKING

The test cases for the search room booking are given in table 5.5

Test number	Test description	Steps to test	Expected results
1	Add validate room details	Add relevant details for each field in the form	Add the details to the booking table as a new row
2		Try to add details with empty fields	Display error messages relevant to each field.
3		Add check in date after check out date	Display error message Checkout cannot be before check in
4	Submit booking details	Click on submit button to submit added details	Display pop up message Added Successfully

Table 5.5: Search room booking test cases

5.2.6 TEST CASES FOR PAYMENT

The test cases for payment are given in table 5.6

Test number	Test description	Steps to test	Expected results
1	Add validate payment details	Add relevant details for each field in the form	Add the details to the payment table as a new row
2		Try to add payment amount with empty field	Display error messages amount should be numeric
3		Enter incorrect payment	Display error message Payment is incorrect
4	Submit payment details	Click on submit button to submit added details	Display pop up message Paid Successfully. Receive an email about payments.

Table 5.6: Payment test cases

5.2.7 TEST CASES FOR ADD ROOM TYPE/RECEPTION HALL

The test cases for add room type/reception hall are given in table 5.7

Test number	Test description	Steps to test	Expected results
1	Add validate room type/reception hall	Add relevant details for each field in the form	Add the details to the room type/ reception

	details		hall table as a new row
2		Try to add price amount with empty field	Display error messages amount should be numeric
3		Enter incorrect price	Display error message amount is incorrect
4	Submit room type/reception hall details	Click on submit button to submit added details	Display pop up message Added Successfully.

Table 5.7: Add room type/reception hall test cases

5.3 USE ACCEPTANCE TEST

Hotel Management System was tested by client side after implementing in client environment. The system was tested by the client to identify the functionalities of the developed system. The system runs in actual environment by feeding data. Some modifications were done according to the client request. After testing the system, it had been requested to test the system using the staff members. With the newly developed system the hotel could function efficiently and smoothly rather than continue using the old method, the client stated. Finally client satisfies with the system and issue client certificate in appendix G.

CHAPTER 6 – CONCLUTION

6.1 INTRODUCTION

Gayana hotel and beach restaurant is a leading hotel in Hambantota district. As a hotel which interact with the customers daily it should have a systematic way to handle information and activities effectively. It is a challenge to handle all the transactions of the hotel and keeping each and every record manually. Therefore, it is a huge necessity to create computer based Hotel Management System to overcome those difficulties for Gayana hotel and beach restaurant.

By interviewing the users, referring the documents of existing manual system and observing the current system were conducted to find out the functional and nonfunctional requirement of the proposed system. The system allowed the user to search room, reception hall, check the availability of resources, services, handle payments and manage user profiles. Also system allowed the administrator to manage user accounts, manage notification, generate reports, handle payments and etc.

The system was implemented using PHP. Apache server used to run the system. Specific design principles were used. MVC pattern is used to separate applications concerns. This Hotel Management System could be helpful for the administrators and users of the hotel functions to have an effective and efficient service.

6.2 FUTURE IMPROVEMENTS

The system can be improved in many ways in the future. Some of the future improvements are given here.

- Make notification through SMS will more user friendly than sending them through email. Therefor suggest making notifications through SMS.
- The system can add more reports according to the client’s current need. Add more effective bar charts and pie charts in report generation module to support top management in decision making.
- Provide online payment facility to the customers to do their transactions via the system using a payment gateway.

6.3 LESSONS LEARNT

As an undergraduate student, the knowledge gained throughout the project was really valuable opportunity to me. I could use the previously learnt theoretical knowledge to perform a task in practical world. It goes beyond simply completing a project for a degree and falls into the category of learning important life lessons as well.

The most useful part of this process was I was able to improve my communication skills by communicating with external parties for gathering information. Also it was great opportunity to identify the current business environment and it broadened my horizons into understanding, how to map those related business processes into a computerized system.

When I developed the system using PHP, MySQL and JavaScript technologies and MVC framework   I could improve my knowledge and collect new techniques that can be used to do our work effectively. Further I could know how to write a report according to given instructions.

REFERENCES

[1]"Software Requirement Specification", *www.tutorialspoint.com*, 2017. [Online]. Available: https://www.tutorialspoint.com/software_testing_dictionary/software_requirement_specification.htm. [Accessed: 08-April, 2017].

[2] quora,*What-is-the-best-online-hotel-management-software*. [Online]Available: https://www.quora.com/What-is-the-best-online-hotel-management-software[Accessed:20-April, 2017]

[3]"5 essential software engineering practices", *TechBeacon*, 2017. [Online]. Available: https://techbeacon.com/5-essential-software-engineering-practices. [Accessed: 01-May- 2017].

[4] Tutorialspoint,*sdlc*. [Online]Available: https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm[Accessed :2-May, 2017]

[5]"SDLC Iterative Model", *www.tutorialspoint.com*, 2017. [Online]. Available: https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm. [Accessed: 8-May, 2017]

[6]"MVC Architecture", *Tutorialsteacher.com*, 2017. [Online]. Available: http://www.tutorialsteacher.com/mvc/mvc-architecture. [Accessed: 10- May, 2017].

[7] Wikipedia, the free encyclopedia,(2017, May 15). *Unified Modeling Language* [Online].Available: https://en.wikipedia.org/wiki/Unified_Modeling_Language[Accessed 15May, 2017]

[8]"GenMyModel", *GenMyModel*, 2017. [Online]. Available: https://www.genmymodel.com/. [Accessed: 25- May- 2017].

[9]"Flowchart Maker & Online Diagram Software", *Draw.io*, 2017. [Online]. Available: https://www.draw.io/. [Accessed: 28- May- 2017].

[10]D. Fadeyev and D. Fadeyev, "10 Useful Techniques To Improve Your User Interface Designs – Smashing Magazine", *Smashingmagazine.com*, 2017. [Online]. Available: https://www.smashingmagazine.com/2008/12/10-useful-techniques-to-improve-your-user-interface-designs/. [Accessed: 05- Jun- 2017].

[11]"XAMPP Installers and Downloads for Apache Friends", *Apachefriends.org*, 2017. [Online]. Available: https://www.apachefriends.org/index.html. [Accessed: 08- jun-2017].

[12]"Welcome to NetBeans", *Netbeans.org*, 2017. [Online]. Available: https://netbeans.org/. [Accessed:15- jun- 2017].

[13]"PHP 7 Tutorial", *www.tutorialspoint.com*, 2017. [Online]. Available: https://www.tutorialspoint.com/php7/. [Accessed: 17- jun- 2017].

[14]"PHPUnit – The PHP Testing Framework", *Phpunit.de*, 2017. [Online]. Available: https://phpunit.de/. [Accessed: 18- jun- 2017].

[15]"MySQL", *Mysql.com*, 2017. [Online]. Available: https://www.mysql.com/. [Accessed: 24- jun- 2017].

[16]"HTML5 Introduction", *W3schools.com*, 2017. [Online]. Available: https://www.w3schools.com/html/html5_intro.asp. [Accessed: 28- jun- 2017].

[17]"CSS Tutorial", *W3schools.com*, 2017. [Online]. Available: https://www.w3schools.com/css/. [Accessed: 28- jun- 2017].

[18]"JavaScript Tutorial", *W3schools.com*, 2017. [Online]. Available: https://www.w3schools.com/js/. [Accessed: 5- jul- 2017].

[19]j. jquery.org, "jQuery", *Jquery.com*, 2017. [Online]. Available: https://jquery.com/. [Accessed: 5- jul- 2017].

[20]"Stack Overflow - Where Developers Learn, Share, & Build Careers", *Stackoverflow.com*, 2017. [Online]. Available: https://stackoverflow.com/. [Accessed: 08- jul- 2017].

[21]"jsPDF", *Mrrio.github.io*, 2017. [Online]. Available: https://mrrio.github.io/. [Accessed: 18- aug- 2017].

[22]"Beautiful HTML5 JavaScript Charts | CanvasJS", *CanvasJS*, 2017. [Online]. Available: https://canvasjs.com/. [Accessed: 25- aug- 2017].

APPENDIX A – SYSTEM DOCUMENTATION

This documentation provides guidelines to the setup of the Hotel Management System. These steps explain hardware and software environment which is need to be installed. Administrator or any other interested party can followed these documentations when installing the system.

HARDWARE REQUIREMENTS

Hardware	Minimum requirement
Processor	Intel Core 2 Duo Processor[2.40 GHz]
Memory	2GB RAM
Hard disk space	Minimum 10GB free disk space or higher
Printer	Dot-matrix, ink jet or laser printer.
Internet	Minimum 512Kbps ADSL connection

Table A.1: hardware requirements

SOFTWARE REQUIREMENTS

Software	Minimum requirement
Operating system	Windows xp, vista, 7 or newer version
Bundle package	XAMPP
Web browser	Internet explorer, Google chrome, safari etc..
Data base server	MySQL server
PDF reader	Adobe Acrobat Reader or any software

Table A.2: software requirements

INSTALLING HOTELMANAGEMENTDB

In order to install the system the necessary software must be installed first. Following is a list of how and where to acquire these software and how to install them.

Installing XAMPP Server

MySQL and phyMyAdmin both provided in XAMPP Server along with other software. It is considerably easier to install XAMPP server than installing its components separately. Therefore it is recommended to download the latest version from https://www.apachefriends.org/download.html and install it on to computer following the instructions.

After the necessary software has been installed, the database for hotel management system hotelmanagementdb must be installed onto the computer. The following is a set of instructions to do so.

Installing the database

- Open localhost by typing the http://localhost/ URL in the browser's address bar. Then home page of XAMPP server should be displayed in browser.
- Click on phpMyAdmin.
- Click on Databases tab in phpMyAdmin home page. Type hotelmanagementdb in the create database text field as database name. Then click on create button. An empty database with the name hotelmanagementdb will be created.
- Next go to the Import tab and click on browse button. Then browse through the supplementary CD's database folder and select hotelmanagementdb.sql file.
- Then click on the go button to import the folder into the created hotelmanagementdb database.

System installation

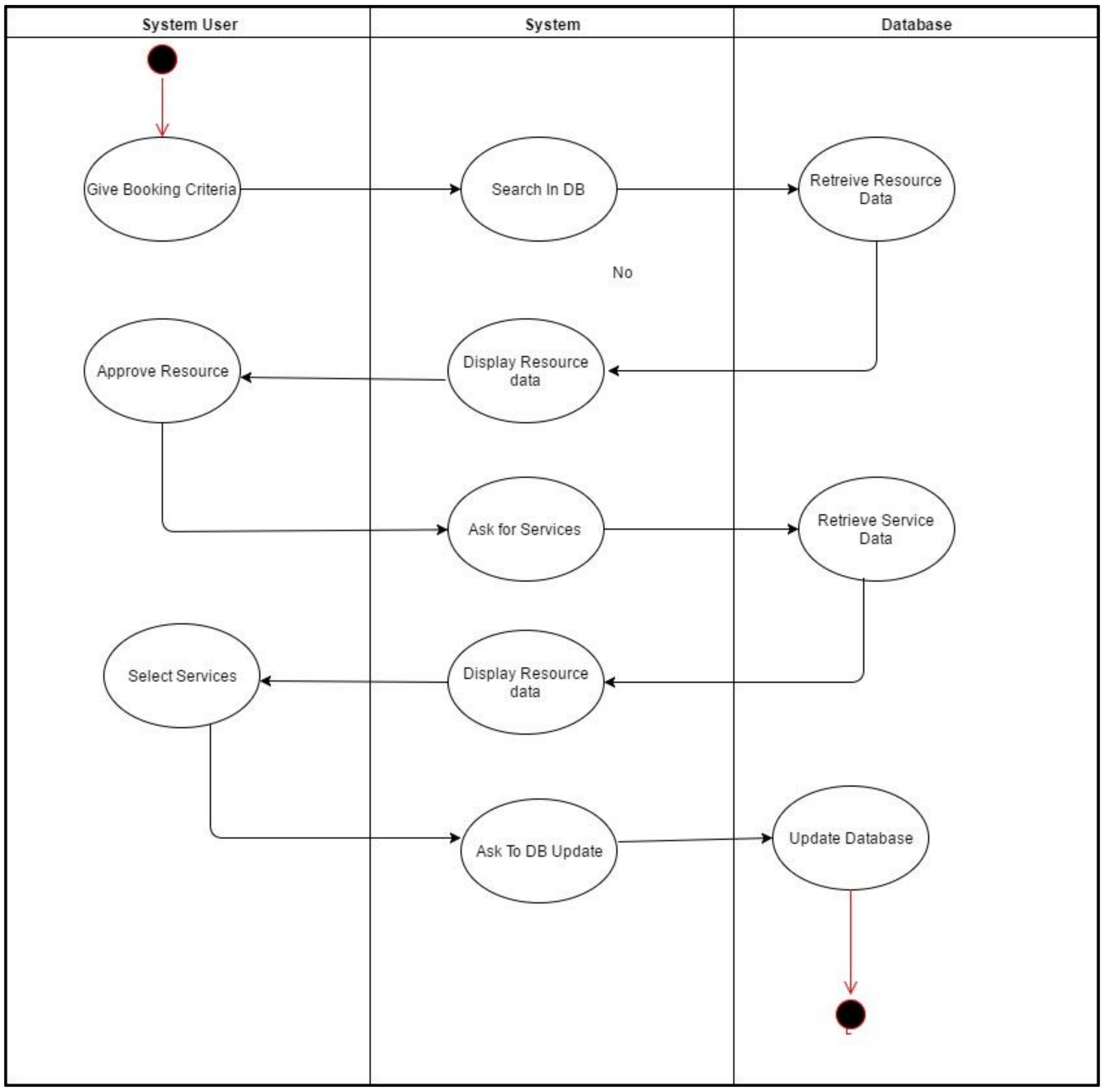
The final step of the installation process is to deploy hotelmanagementdb.
To deploy hotelmanagementdb copy the Hotel_Management_System folder given in the supplementary CD and paste it inside the htdocs folder in the C:\xampp\htdocs

After all the installations completed open a web browser and type http://localhost/Hotel_Management_System/ in address bar to view the home page. By typing a valid username and password combination can gain access to the system.

APPENDIX B – DESIGN DOCUMENTATION

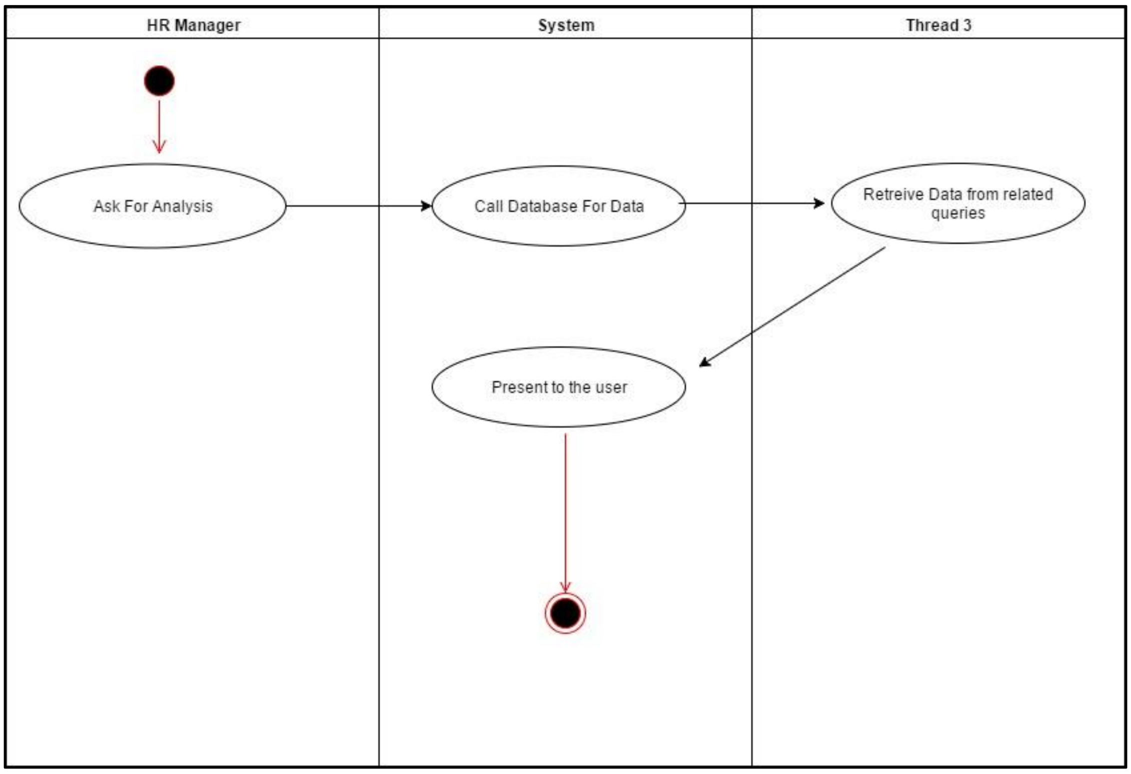
Activity diagrams

1. Activity diagram for Booking



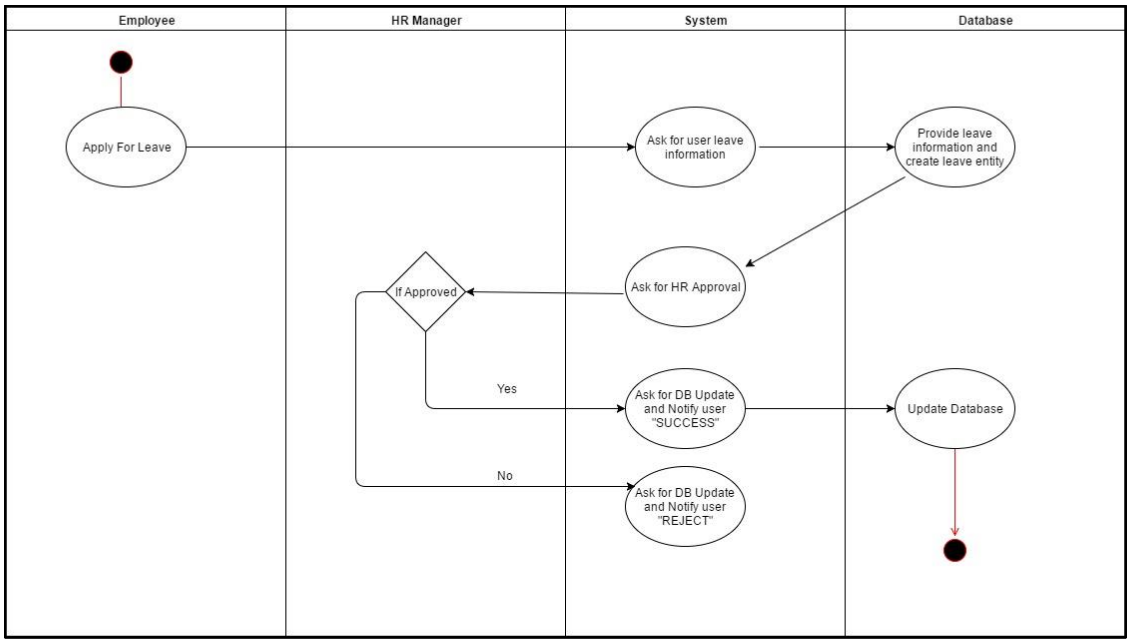
FigureB.1: Activity diagram for Booking

2. Activity diagram for Analysis



FigureB.2: Activity diagram for Analysis

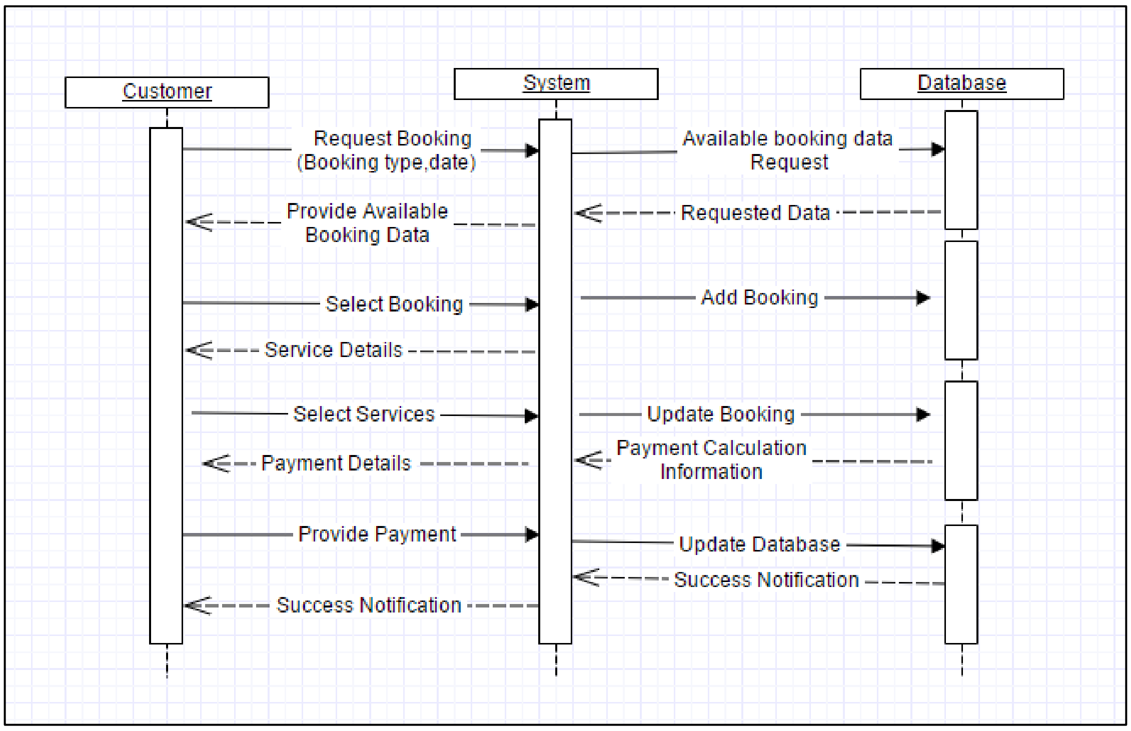
3. Activity diagram for Leaves


Employee
HR Manager
System
Database
Apply For Leave
Ask for user leave information
Provide leave information and create leave entity
Ask for HR Approval
If Approved
Yes
No
Ask for DB Update and Notify user "SUCCESS"
Ask for DB Update and Notify user "REJECT"
Update Database

FigureB.3: Activity diagram for Leaves

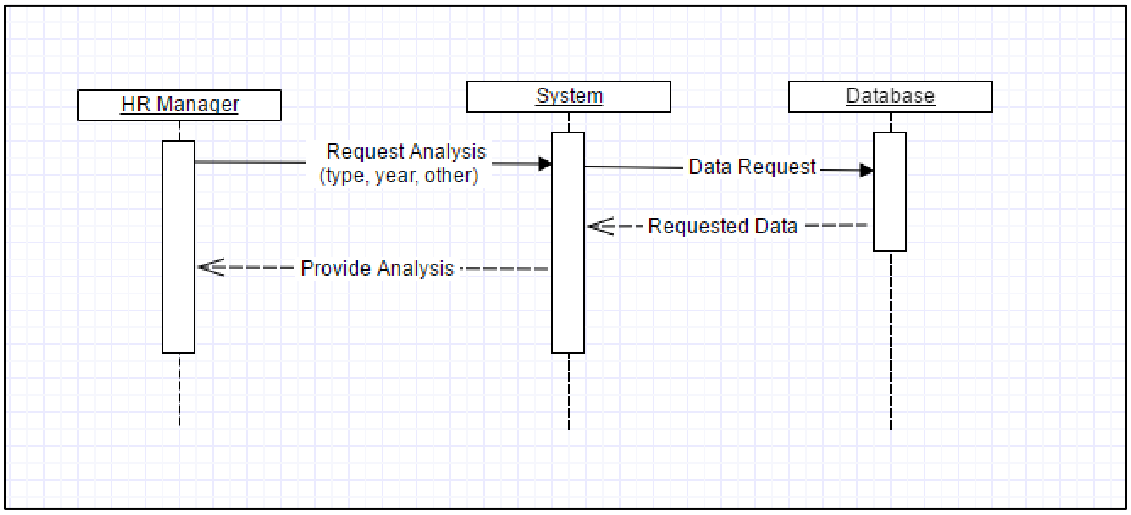
Sequence diagrams

1. Sequence diagram for Booking



FigureB.4: Sequence diagram for Booking

2. Sequence diagram for Analysis

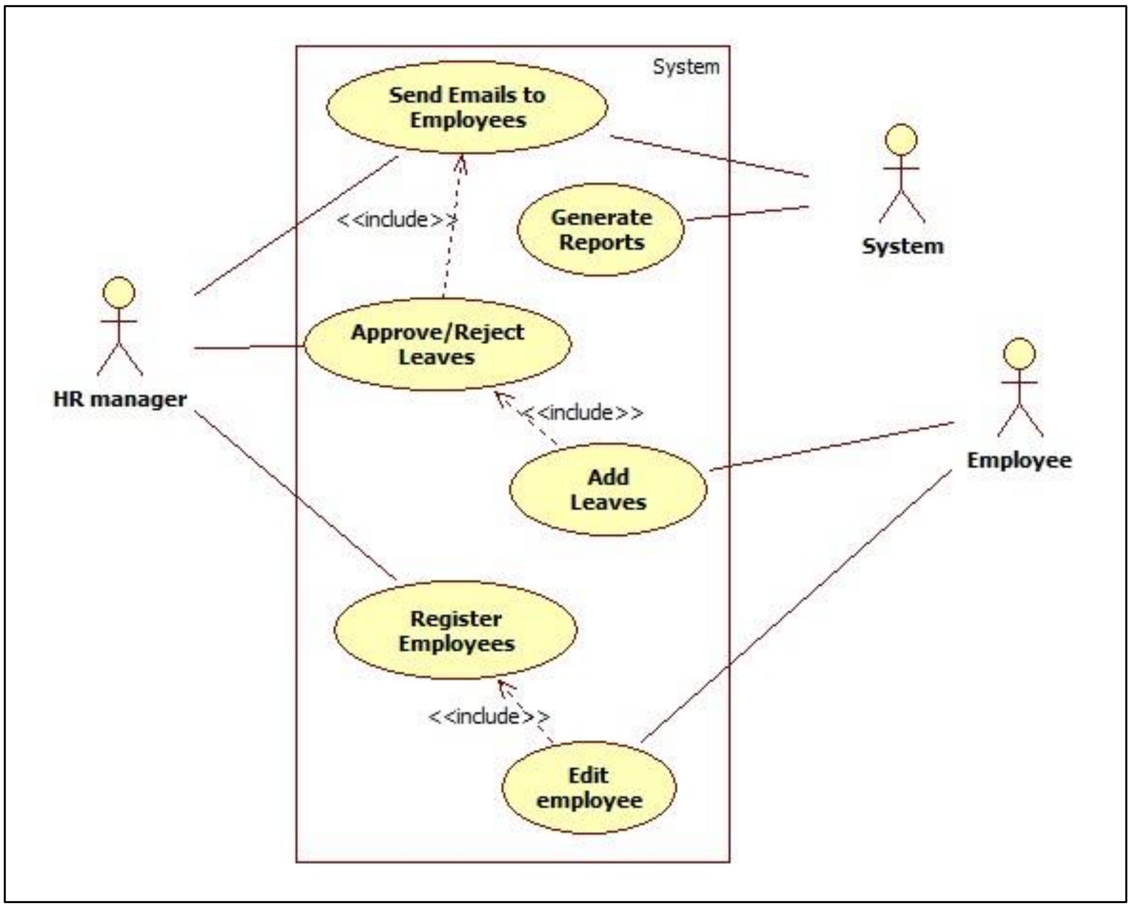


FigureB.5: Sequence diagram for Analysis

Use case diagrams with detailed information

Some of detailed use case diagrams and relative narratives are given here to explain the use case diagram shown in chapter 3.

1. Use case diagram for HR



FigureB.6: use case diagram of HR

Use case	Register employees
Actors	HR manager
Description	
Register employees	
Precondition	
HR manager should login to the system using system user account	
Flow of events	
• If a new employee add employee details and submit. • If existing employee edit employee details and submit	
Post conditions	
Employee details should be saved in database.	

Table B.1: Use case description for register employee

Use case	Edit Employee
Actors	HR manager, Employee
Description	
Edit Employee details	
Precondition	
Employee should be added by HR manager. HR manager should login to the system using system user account. Employee should login to the system using employee user account	
Flow of events	
• Edit employee details and submit	
Post conditions	
Employee details should be updated and saved in database.	

Table B.2: Use case description for edit employee

Use case	Add Leaves
Actors	Employee
Description	
Add leaves and check leave request status	
Precondition	
Employee should login to the system using employee user account	
Flow of events	
• Select date and leave type • State description and submit request • Check whether requested leaves approved or rejected	
Post conditions	
Requested leaves should be submitted to HR manager consideration.	

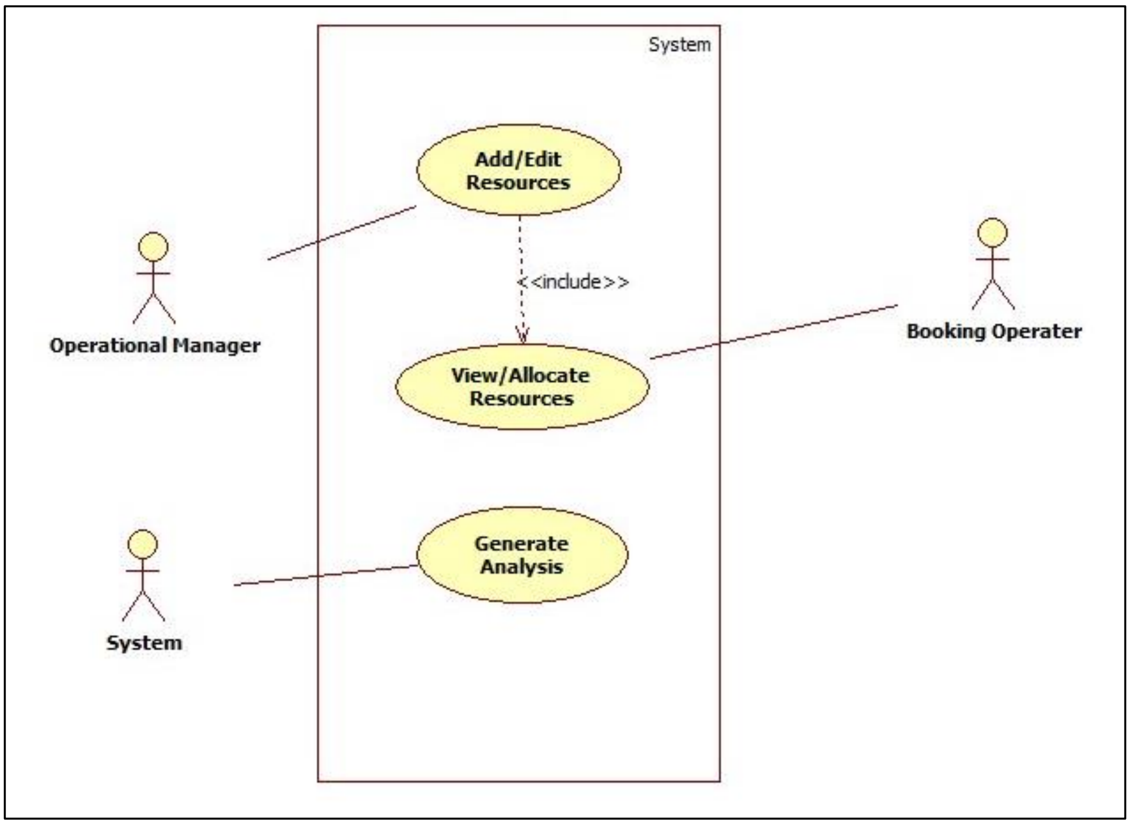
Table B.3: Use case description for add leaves

Use case	Approve/reject leaves
Actors	HR manager, System
Description	
Check leaves and approve or reject	
Precondition	
HR manager should login to the system using system user account	
Flow of events	
• Load the leave request and check them. • Click on approve or reject and submit.	

Post conditions
Leave request approval or reject status should be saved and updated in database. System sends email notifications to employee.

Table B.4: Use case description for approve leaves

2. Use case diagram for resource module

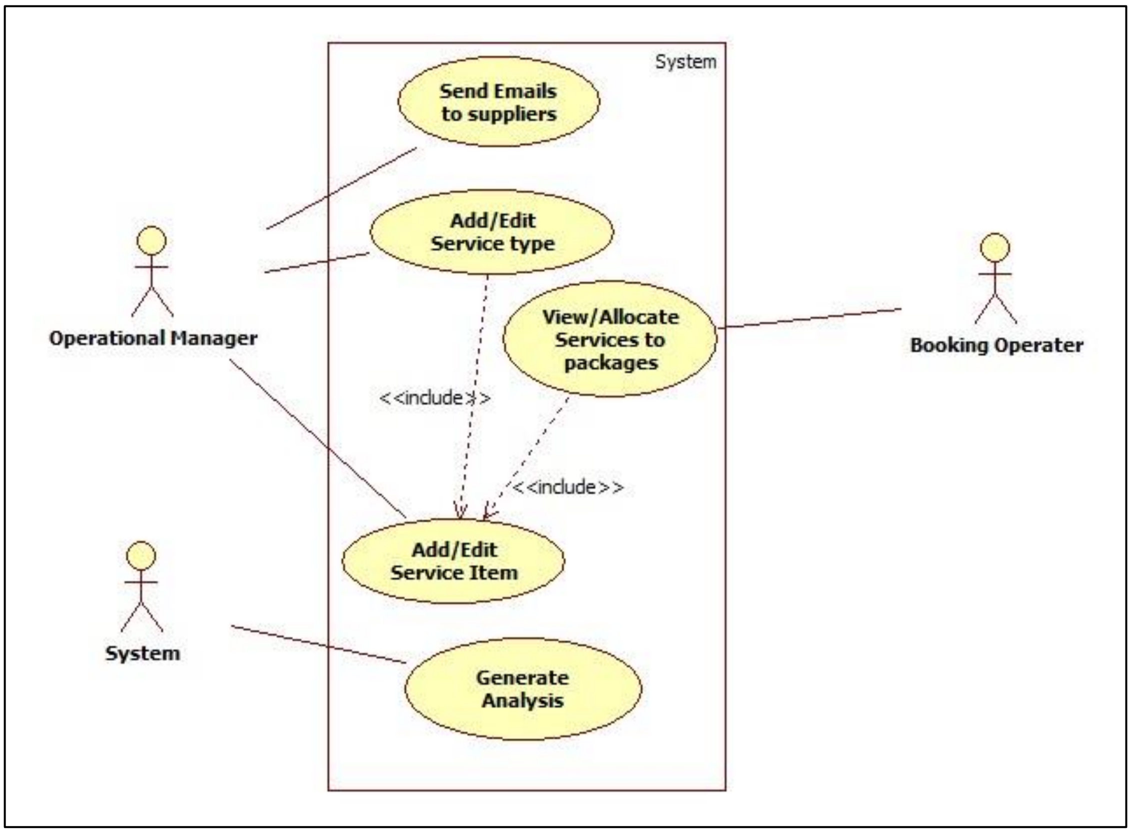


FigureB.7: use case diagram of resource

Use case	Add/Edit resources
Actors	Operational manager
Description	
Add rooms and reception halls	
Precondition	
Operational manager should login to the system using system user account	
Flow of events	
• Add rooms or reception halls details • Click on submit.	
Post conditions	
Resource details updated in the database. Booking operator gets a chance to allocate and view resources.	

Table B.5: Use case description for add/edit resource

3. Use case diagram for service module



FigureB.8: use case diagram of service

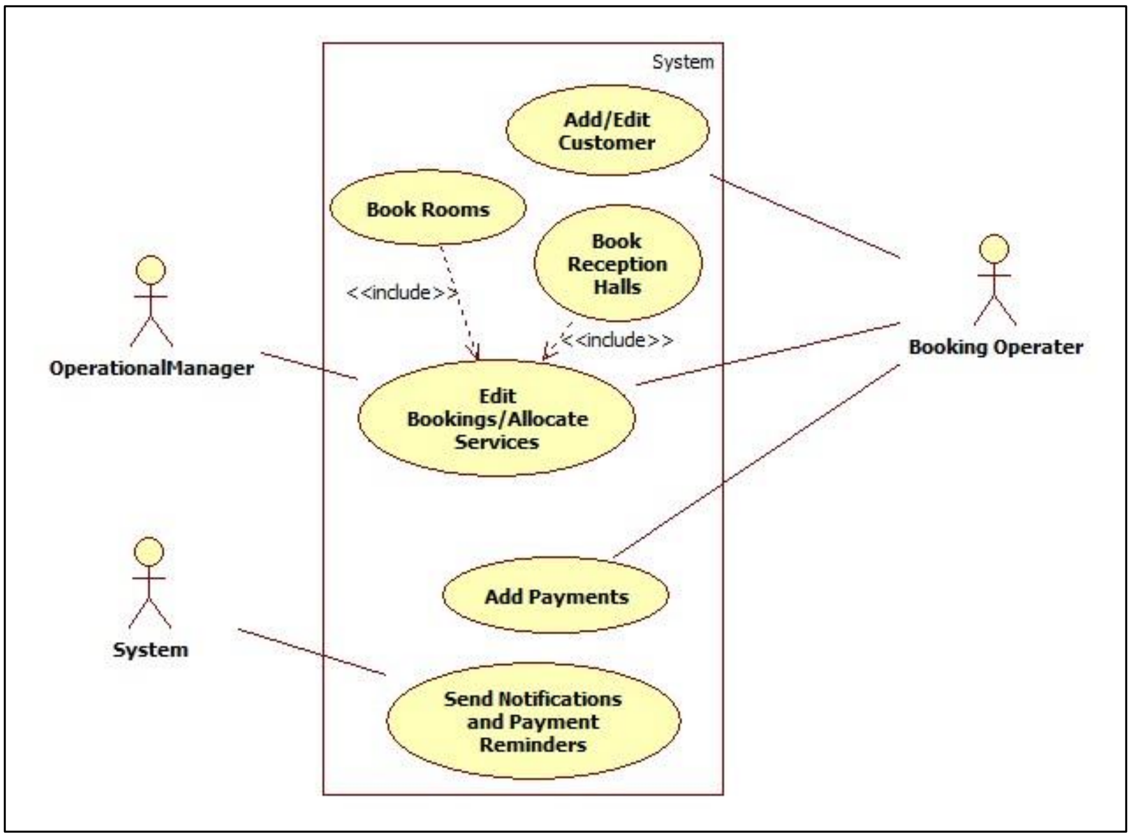
Use case	Add/Edit service item
Actors	Operational manager
Description	
Add/edit service items to allocate services to packages	
Precondition	
Service type should be added. Booking operator should login to the system using system user account	
Flow of events	
• Select service type • Add service items • Click on submit.	
Post conditions	
Service items details updated in the database. Booking operator gets a chance to allocate and view services.	

Table B.6: Use case description for Add/edit service

Use case	View/allocate services
Actors	Booking operator
Description	
Add/edit service items to allocate services to packages	
Precondition	
Service type and item should be added. Operational manager should login to the system using system user account	
Flow of events	
• Select package • Add services to the package • Submit.	
Post conditions	
Service item and booking details updated in the database.	

Table B.7: Use case description for view/allocate service

4. Use case diagram for package module



FigureB.9: use case diagram of package module

Use case	Edit bookings/allocate services
Actors	Booking operator
Description	
Edit bookings or allocate services to packages	
Precondition	
Service type and item should be added. Resources should add. Customer should be registered. Operational manager should login to the system using system user account	
Flow of events	
• Allocate/book resources • Add services to the package • submit	
Post conditions	
Submit package details to the system and generate payment details.	

Table B.8: Use case description for edit bookings/allocate services

Use case	Add Payment
Actors	Booking operator, system
Description	
Add payments and send notifications	
Precondition	
Operational manager should login to the system using system user account. Customer should book a package.	
Flow of events	
• calculate and generate payment details according to the package • Add payments to the system • submit	
Post conditions	
Submit payment details to the system. System sends notification emails to the customer.	

Table B.9: Use case description for payments

APPENDIX C – USER DOCUMENTATION

This document is for the reference of the users who willing to access the Hotel Management System. Users should know how to navigate within the system and how to use the functionalities of the system.

The users are classified as System Use and Employee. Different privileges are assigning to different user roles.

Login interface

Following is the login screen (Figure C.1) appears at the startup. Providing correct employee id and password users can login to the system.

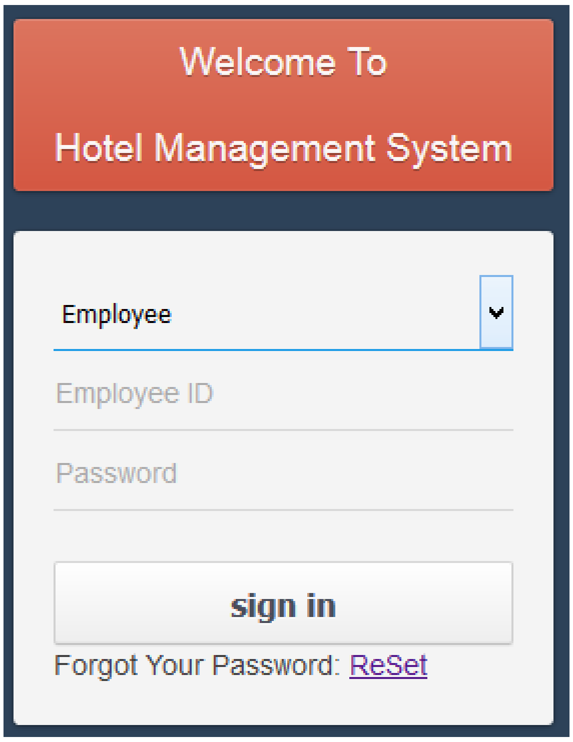
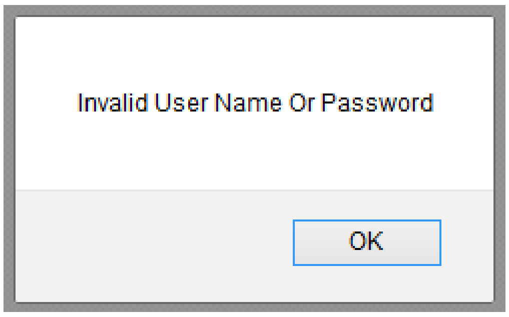


Figure C.1: Login screen

System checks logging details. If user has entered invalid details user will not allowed to enter to the system and display error messages. (Figure C.2)


Invalid User Name Or Password

OK

Figure C.2: Error message of invalid login details

When user forgot his or her password can click on reset link (Figure C.3) and request for a new password providing necessary details.

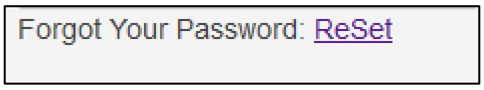
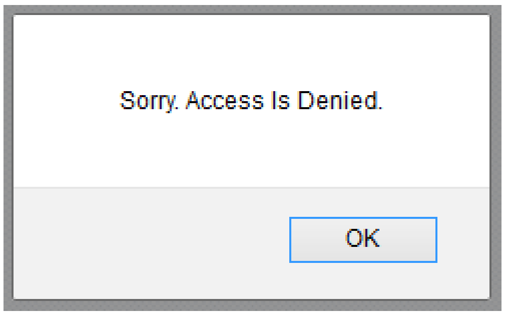

Forgot Your Password: ReSet

Figure C.3: Reset link

According to the user type user get the chance to access the system. User privileges control by different controller. User get error message (Figure C.4) when enter to non-privilege function.


Sorry. Access Is Denied.

OK

Figure C.4: Error Message

Sign in and Sign Out from the system

When user log in to the system sign in details shown in figure C.5

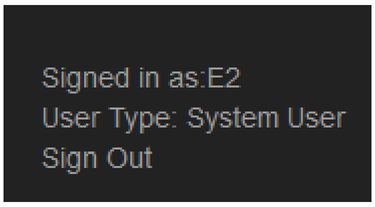

Signed in as:E2
User Type: System User
Sign Out

Figure C.5: Sign in details

After clicking on sign out user can successfully logout from the system and it will direct user back to the login page.

Main menu

Once logged in to the system user is directed to the home page. All the functions of the system appear in the main menu. Main menu changes according to the user type. Main menu for the system user login shown in figure C.6

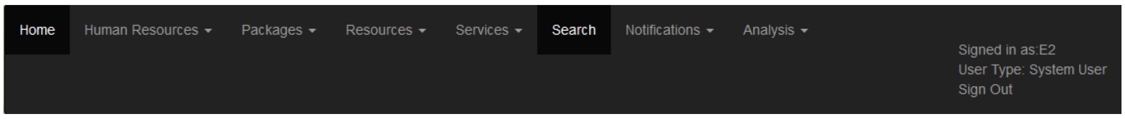
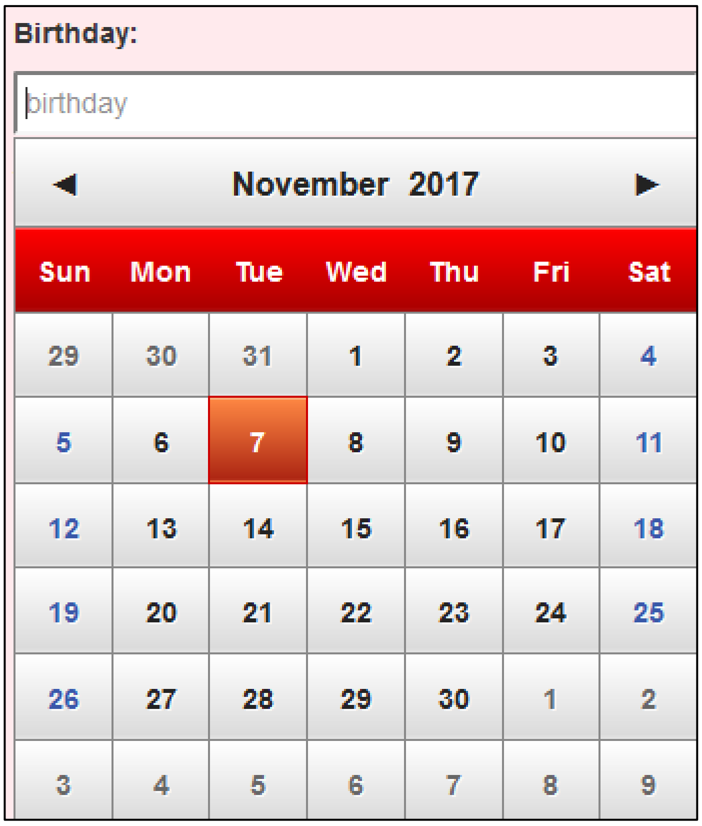


Figure C.6: Main menu

Date picker - The date needed by the user must click to insert. (Figure C.7)

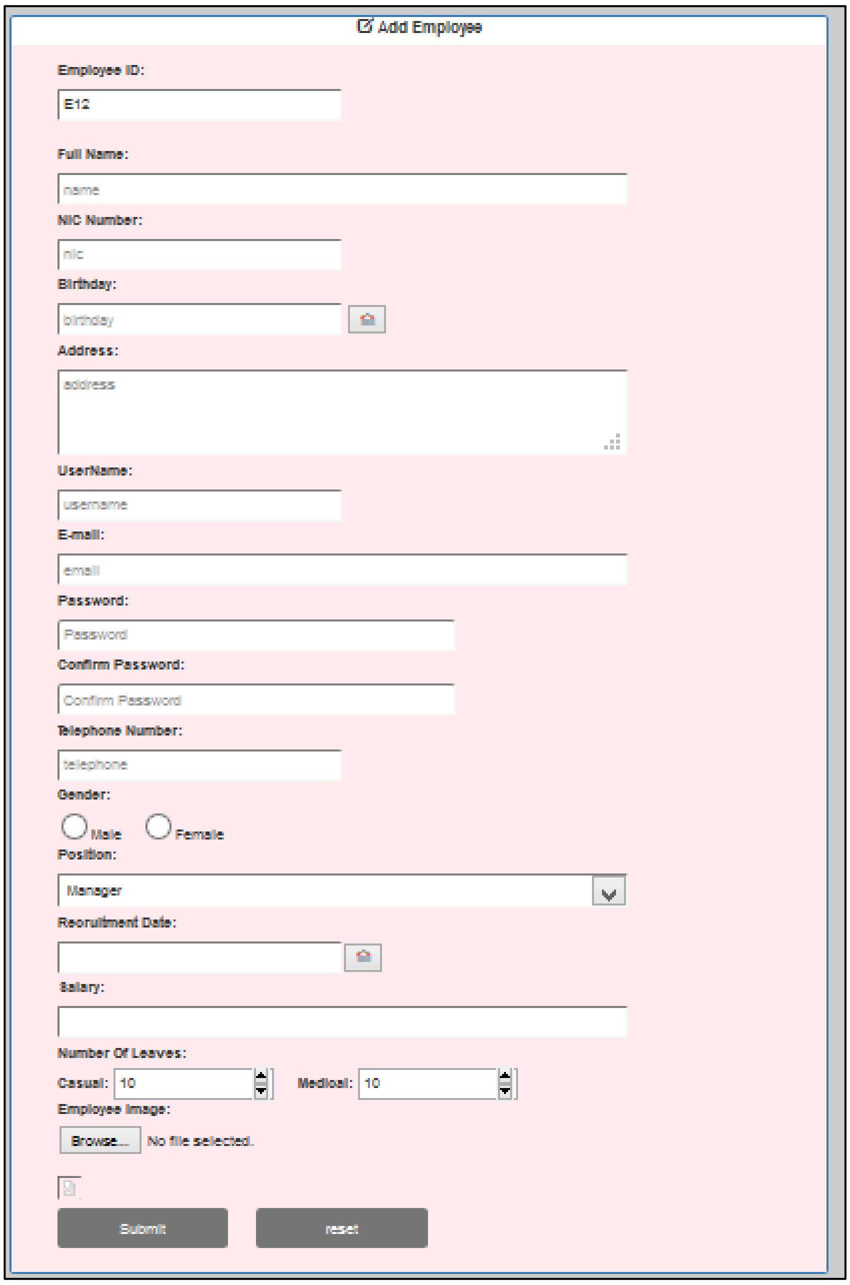


29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

Figure C.7: Calender

Add Employee Interface

By clicking on add employee link in the Human resources menu system user can add a employee to the system. (Figure C.8) Only authorized system users can add employee. After filling relevant information in the form click on submit button. If the member has been added successfully then system will generate a message (Figure C.9) or if not successful display any error message.



Add Employee

Employee ID: E12

Full Name: name

NIC Number: nic

Birthday: birthday

Address: address

UserName: username

E-mail: email

Password: Password

Confirm Password: Confirm Password

Telephone Number: telephone

Gender: ◯ Male ◯ Female

Position: Manager

Recruitment Date:

Salary:

Number Of Leaves:

Casual: 10 Medical: 10

Employee Image:

Browse... No file selected.

Figure C.8: Add employee interface

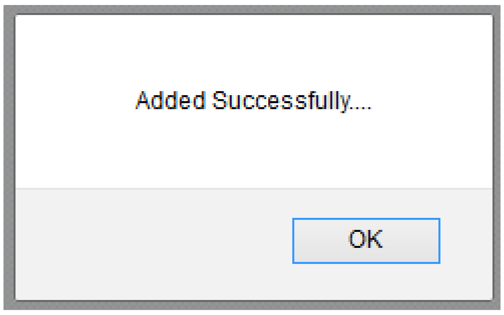

Added Successfully....

Figure C.9: Successful message

Edit Employee

Authorized system user can edit employee details using edit employee link in Human resources menu. Then employee details loaded to the edit employee form. After edited successfully then system will generate a message (Figure C.10) Employee also can edit self profile using manage your details link in Human resources menu.

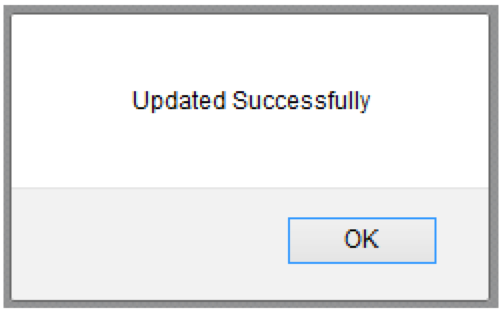
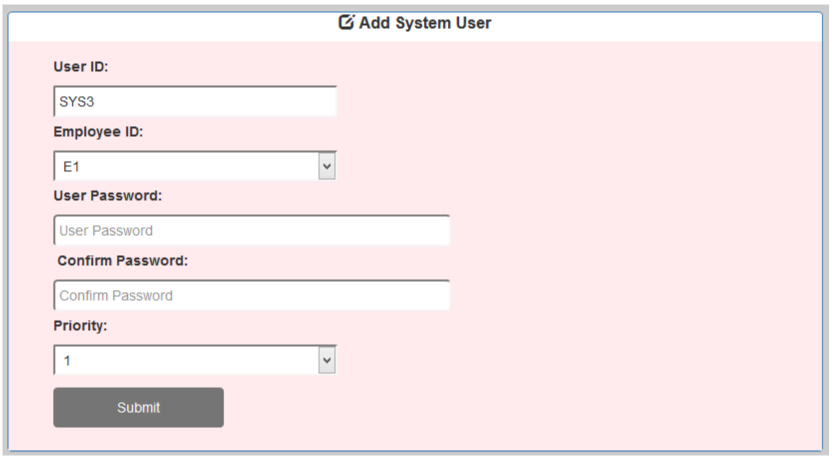

Updated Successfully

Figure C.10: Updated successful message

Add System User Interface

Authorized system user can create system users filling and submitting add system user interface (Figure C.11) under Human resources menu.


Add System User

User ID: SYS3

Employee ID: E1

User Password: User Password

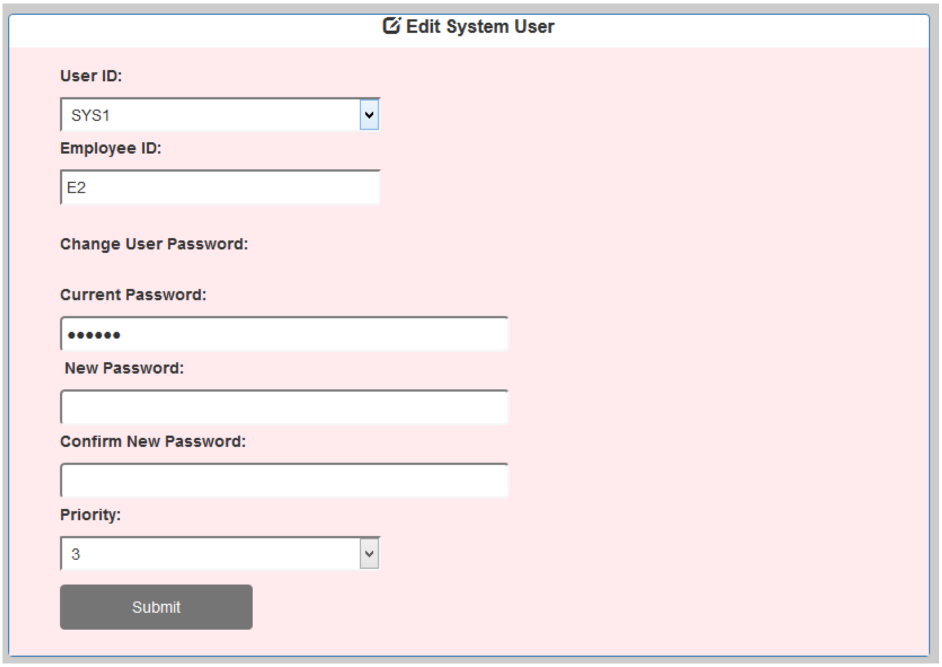
Confirm Password: Confirm Password

Priority: 1

Figure C.11: Add system user interface

Add System User Interface

Authorized system user can edit system user details using edit system user interface (Figure C.12) under Human resources menu. Then the details will load to the form.



Edit System User

User ID: SYS1

Employee ID: E2

Change User Password:

Current Password: ••••••

New Password:

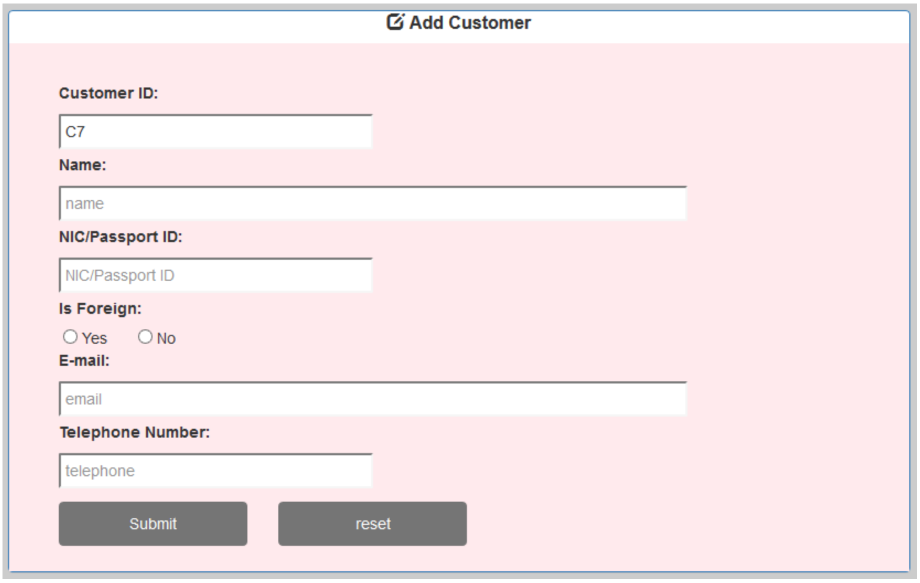
Confirm New Password:

Priority: 3

Figure C.12: Edit system user interface

Add Customer

New customer can be added using add customer interface (Figure C.13) under package menu. Before make a booking every customer should register in the system. Bookings can be done using customer ID.



Add Customer

Customer ID: C7

Name: name

NIC/Passport ID: NIC/Passport ID

Is Foreign: ○ Yes ○ No

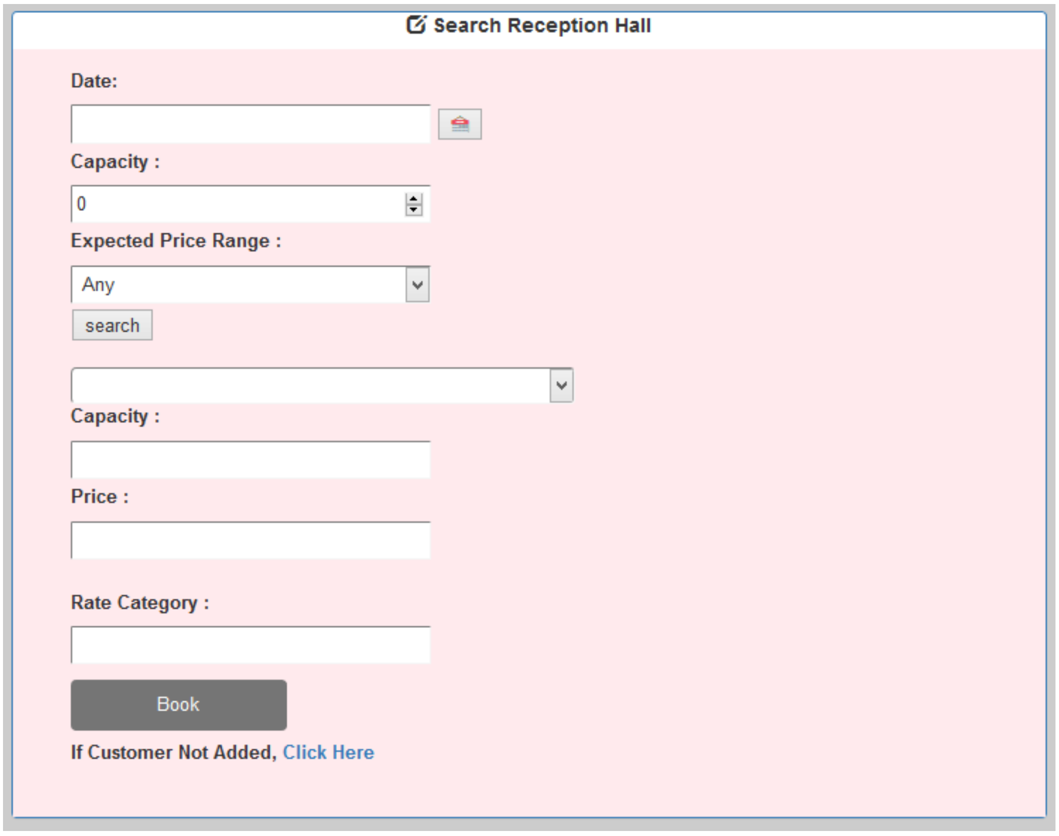
E-mail: email

Telephone Number: telephone

Figure C.13: Add customer interface

Search Reception hall / add reception hall booking

Authorized system user can search reception hall for a booking. (Figure C.14) But only registered customers can make a booking. Before making a booking customer should register using the add customer link under package menu or following the link under search reception hall.


Search Reception Hall

Date:

Capacity : 0

Expected Price Range : Any

search

Capacity :

Price :

Rate Category :

Book

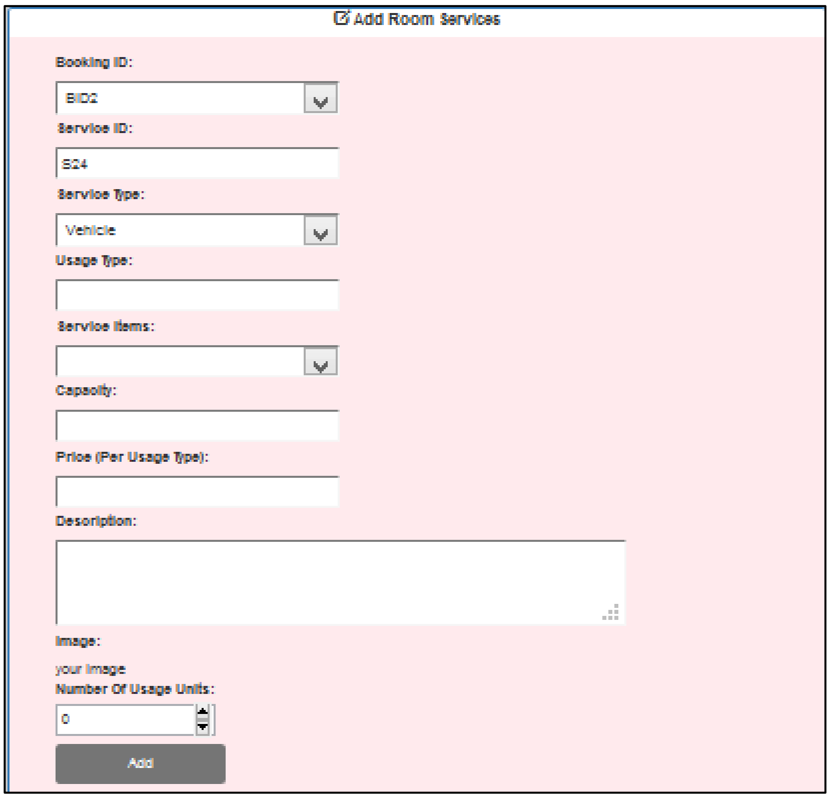
If Customer Not Added, Click Here

Figure C.14: Edit system user interface

- Add room booking also same as add reception hall booking.

Add room services

Room services can be added after making a room booking. To add room services use add room service interface (Figure C.15) under package menu.


Add Room Services

Booking ID:

BID2

Service ID:

S24

Service Type:

Vehicle

Usage Type:

Service Items:

Capacity:

Price (Per Usage Type):

Description:

Image:

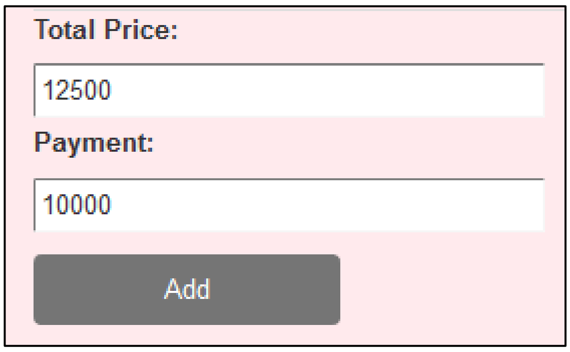
your image

Number Of Usage Units:

0

Figure C.15: Add room services user interface

After adding room services total price calculate and display in the same window as figure C.16 Customer can do payment according to total price.


Total Price:

12500

Payment:

10000

Figure C.16: Payment details

If customer needs some other services after his booking system user can use extra services link under package menu to add extra services. Using customer id can load booking details. If system user adds already added service to the package error message display (figure C.17) and prevent that error.

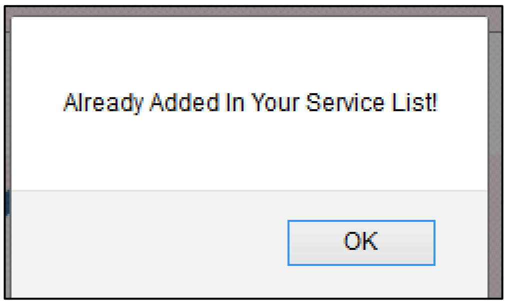
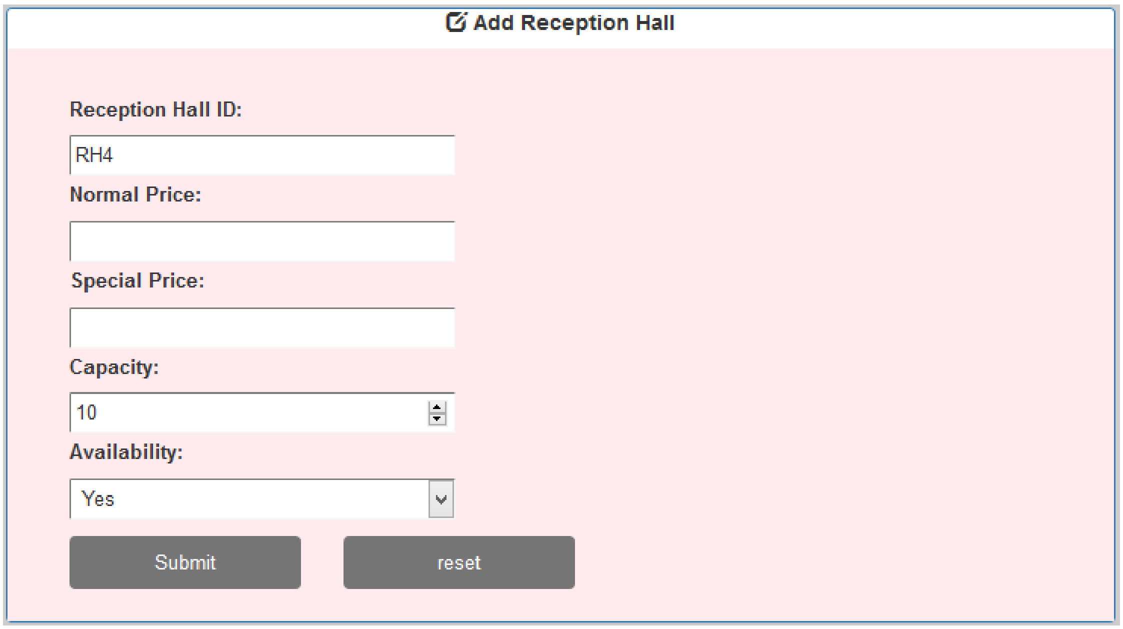

Already Added In Your Service List!

Figure C.17: Error message

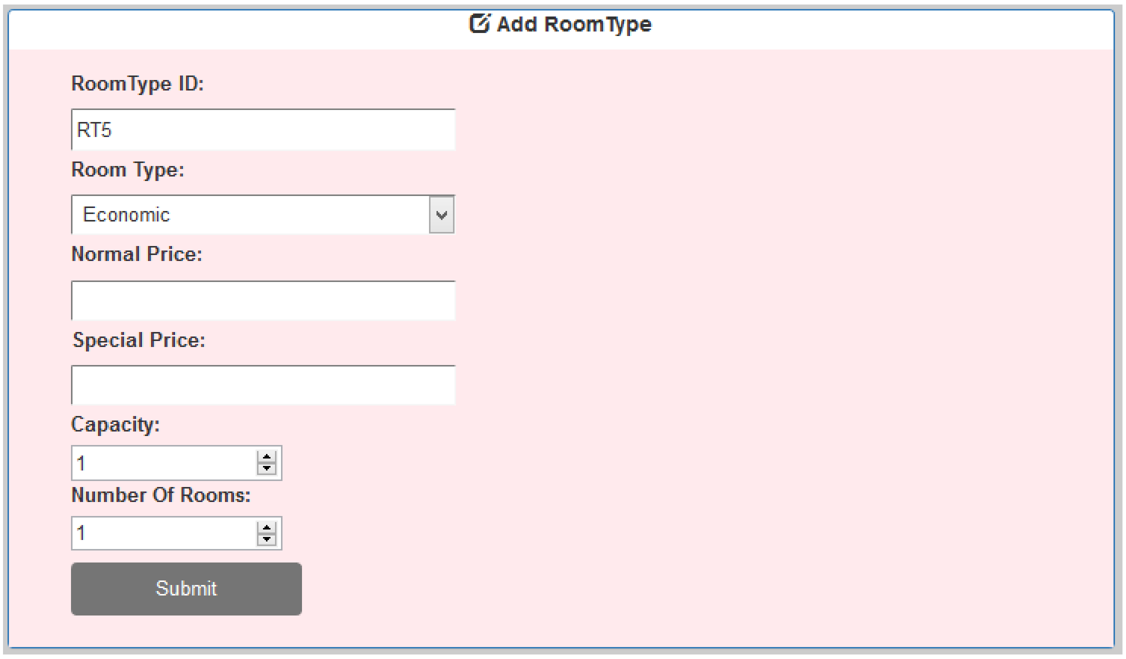
Add reception Hall/ Add Room type

Authorized system user can add resources such as reception hall (Figure C.18) and room type (Figure C.19) using interfaces under resource menu links. Edit functions can be done same as following edit links.


Add Reception Hall

Reception Hall ID:	RH4
Normal Price:	
Special Price:	
Capacity:	10
Availability:	Yes

Figure C.18: Add reception hall


Add RoomType

RoomType ID:
RT5
Room Type:
Economic
Normal Price:

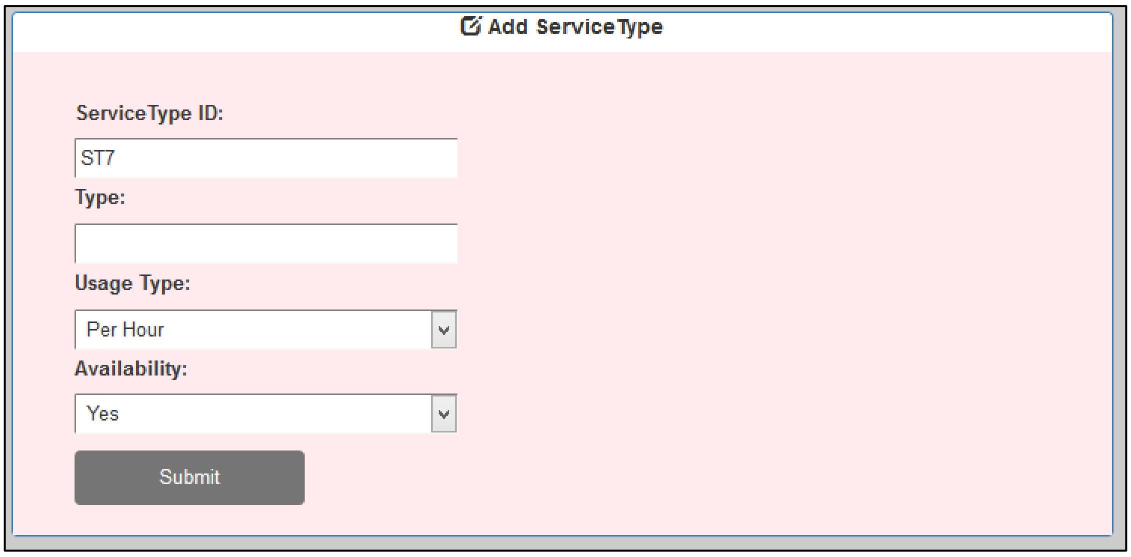
Special Price:

Capacity:
1
Number Of Rooms:
1

Figure C.19: Add room type

Add service type /Add service item

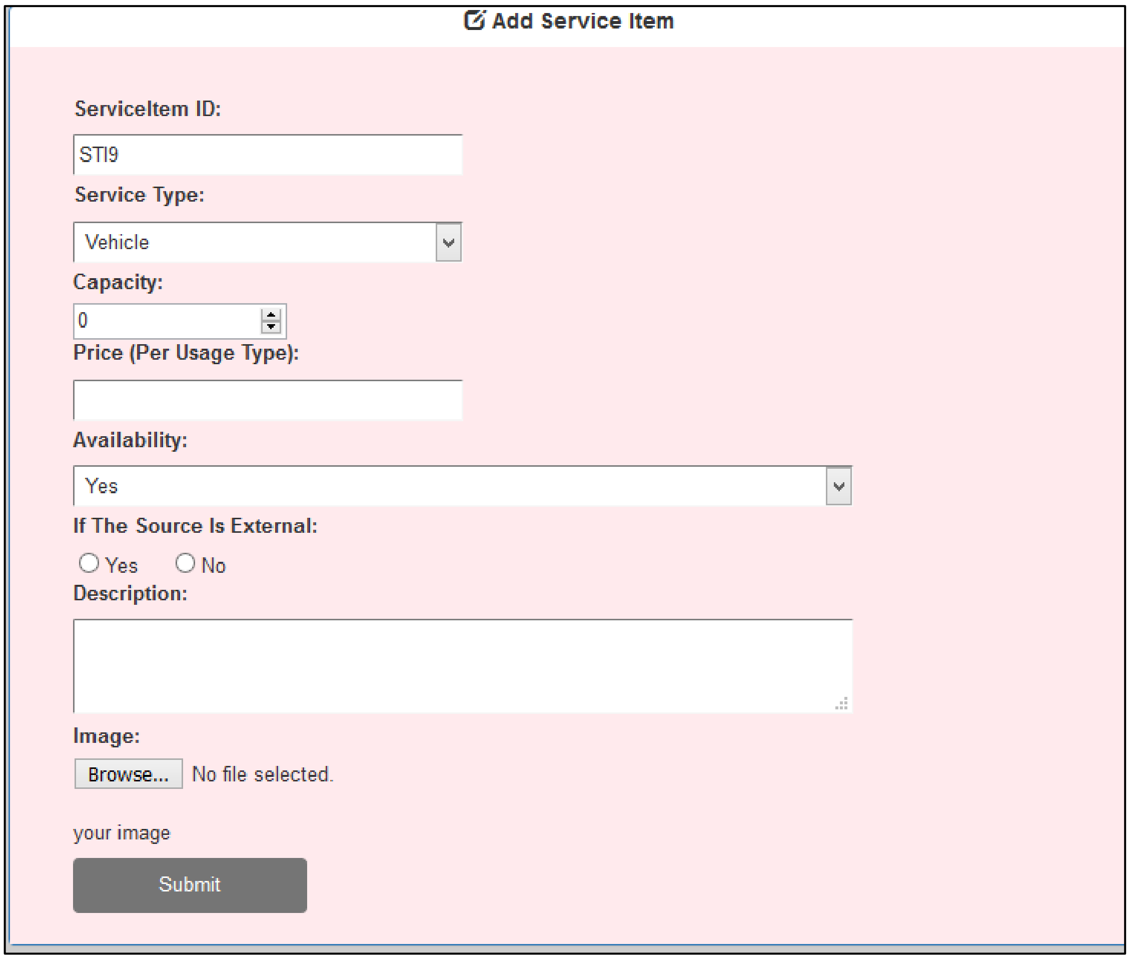
Authorized system user can add service types (Figure C.20) and service items (Figure C.21) using links under service menu. Edit functions can be done same as following edit links.


Add ServiceType

ServiceType ID:
ST7
Type:

Usage Type:
Per Hour
Availability:
Yes

Figure C.20: Add service type


Add Service Item

ServiceItem ID:

STI9

Service Type:

Vehicle

Capacity:

0

Price (Per Usage Type):

Availability:

Yes

If The Source Is External:

Yes No

Description:

Image:

Browse... No file selected.

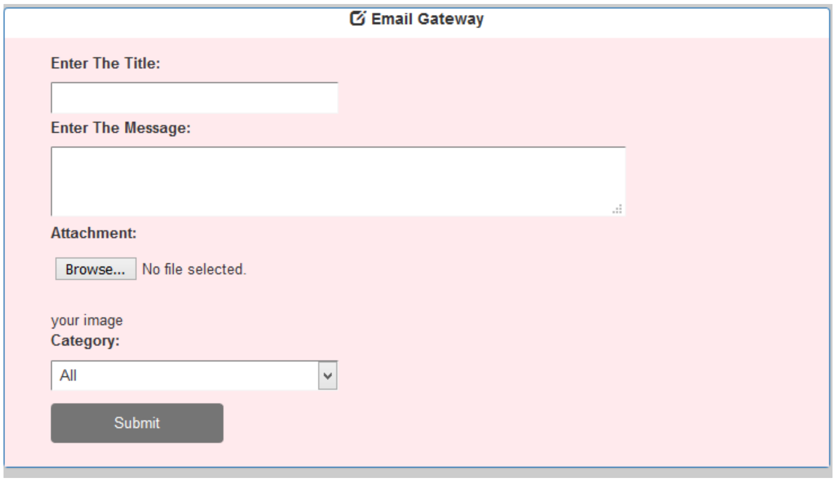
your image

Submit

Figure C.21: Add service item

Email notifications

Authorized system user can send email notifications to employers and customers using email gateway (Figure C.22)


Email Gateway

Enter The Title:

Enter The Message:

Attachment:

Browse... No file selected.

your image

Category:

All

Submit

Figure C.22: Email gateway interface

Analysis

Authorized system user can do several analyses such as service, booking, customer and leave details under analysis menu. Service analysis interface in figure C.23

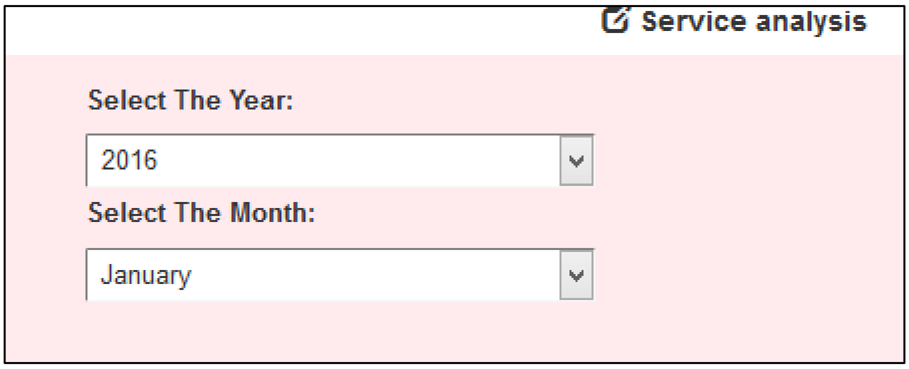


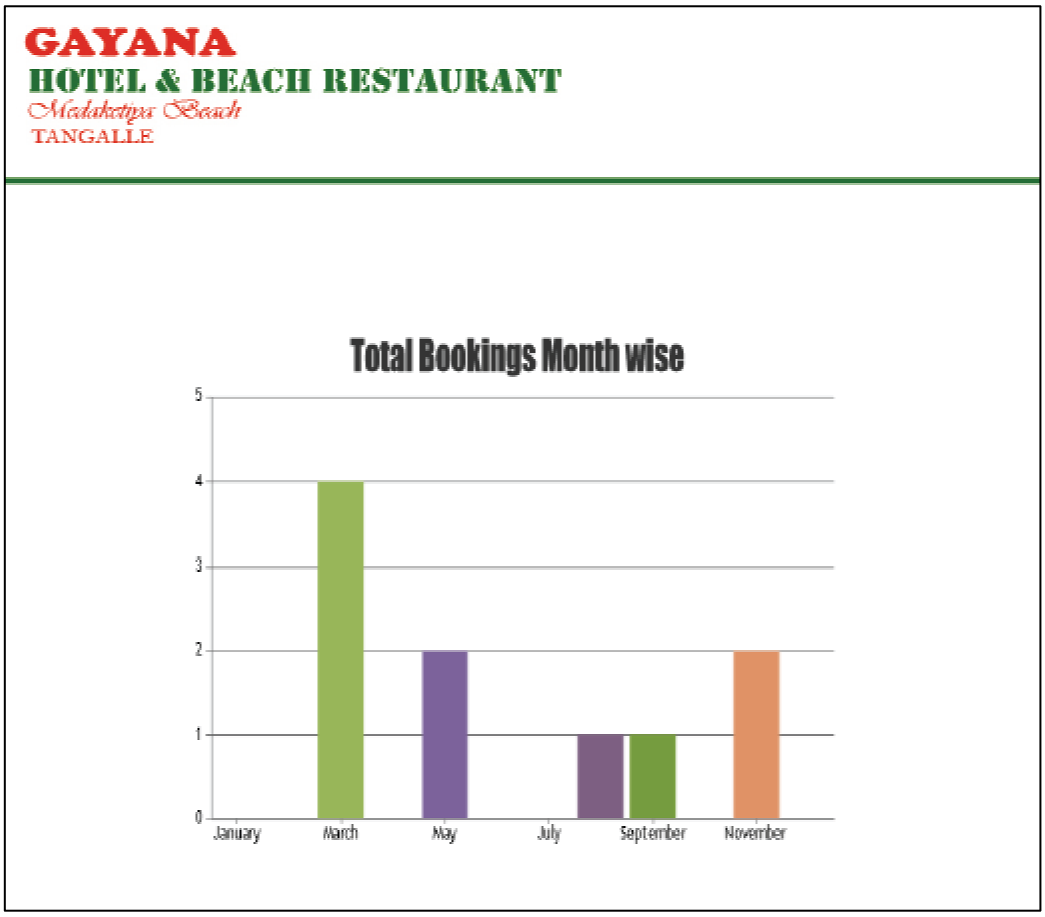
Figure C.23: Service analysis interface

APPENDIX D – MANAGEMENT REPORT

Report

- **Total bookings month wise**

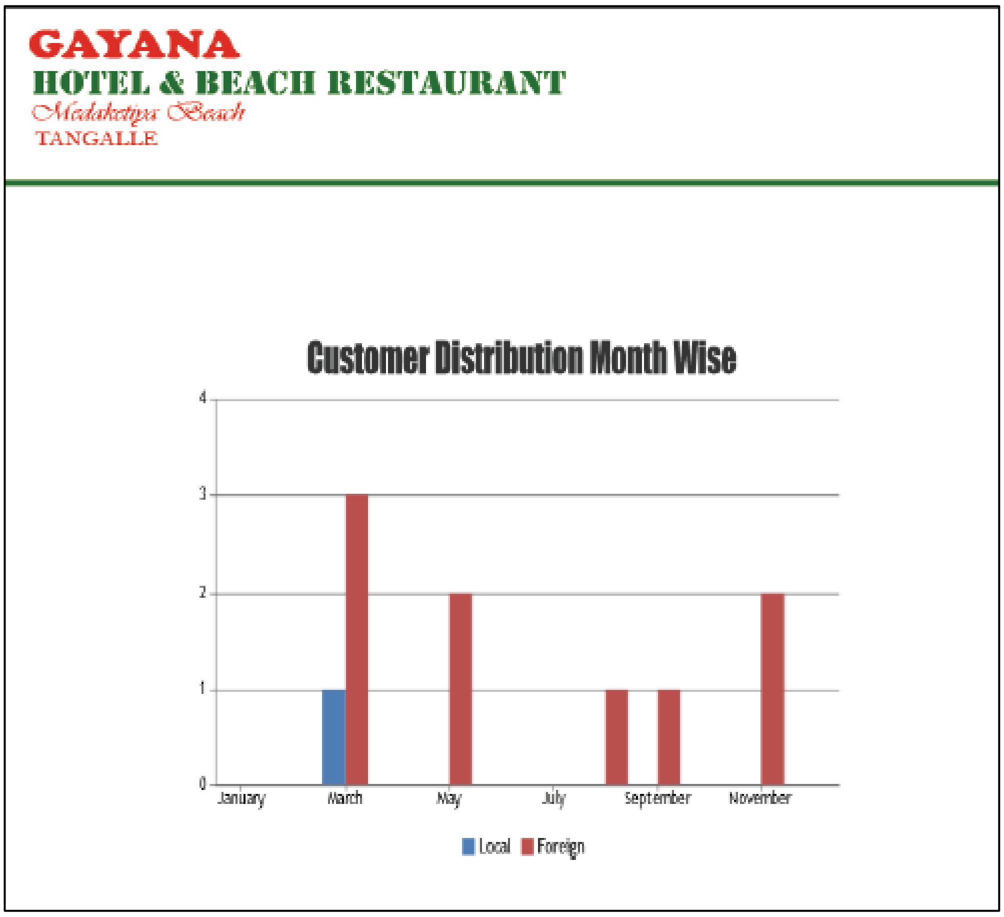
Managers can get details of bookings month wise by using this report. Month wise booking report of the hotel shown in figure D.1



FigureD.1: Report of total booking

- **Customer distribution month wise**

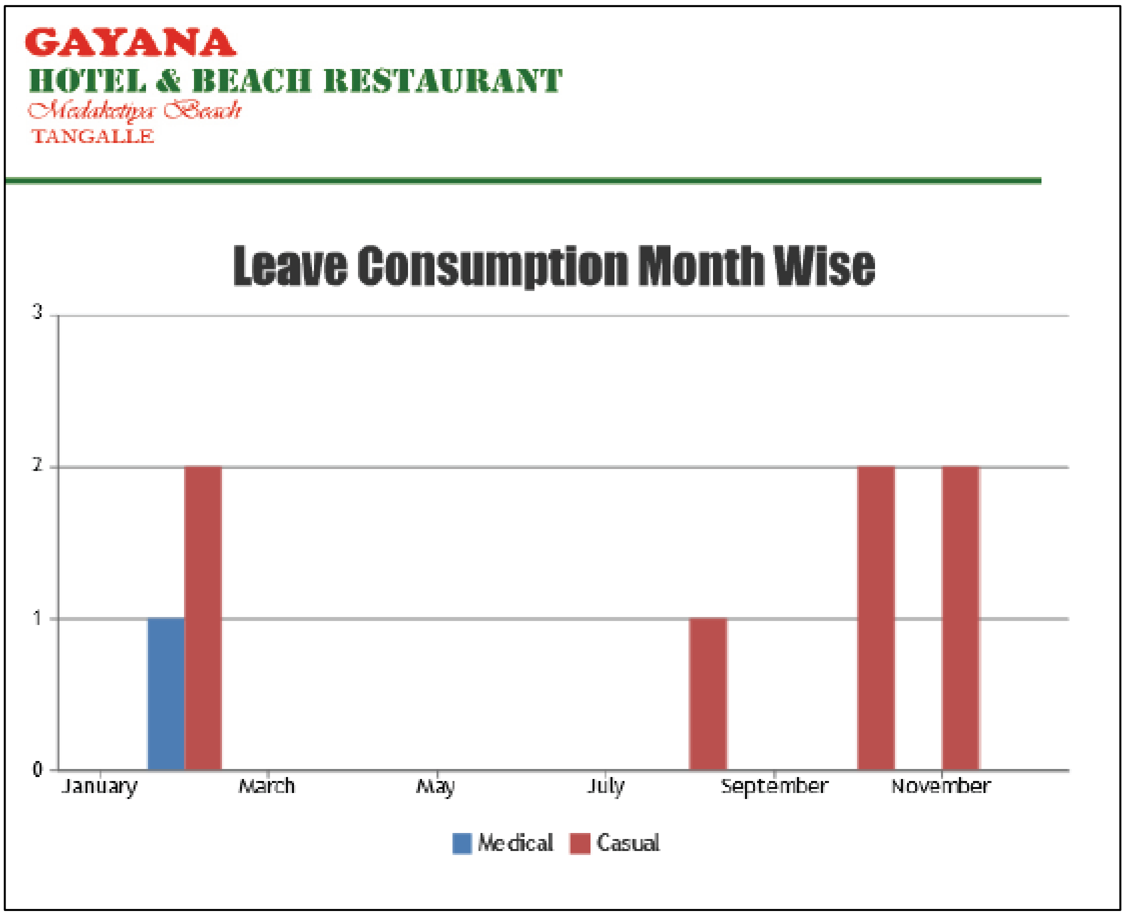
Managers can get details of customers' month wise by using this report. Month wise customer distribution report of the hotel shown in figure D.2



FigureD.2: Report of customer distribution

- **Employee leave analysis**

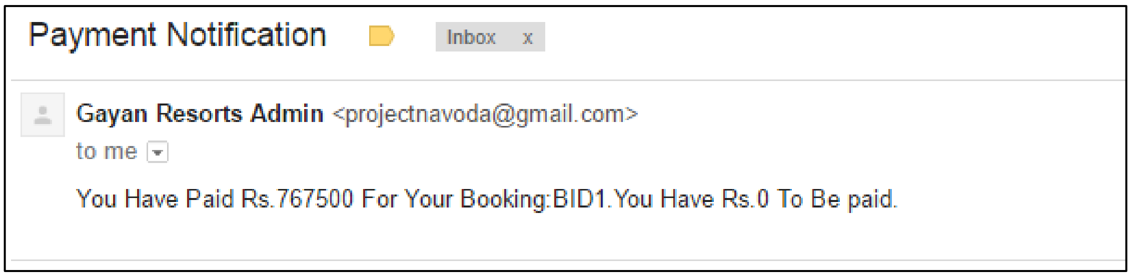
Managers can get details of employee leaves by using this report. Month wise employee leaves report of the hotel shown in figure D.3



FigureD.3: Report of employee leaves

Email Notifications about payments.

Customers get email notifications about their payments. Email notifications shown in figure D.4


Payment Notification Inbox x

Gayan Resorts Admin <projectnavoda@gmail.com>
to me

You Have Paid Rs.767500 For Your Booking:BID1.You Have Rs.0 To Be paid.

FigureD.4: Email notification

APPENDIX E – TEST RESULT

Major test cases of the evaluation stage are given here with test results.

Test result for login authentication

The test result for the user login authentication are given in table E.1

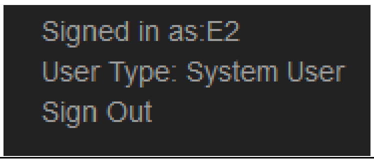
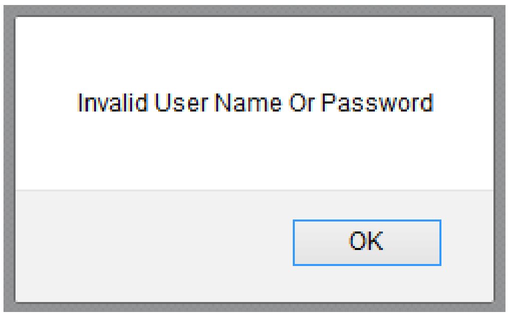
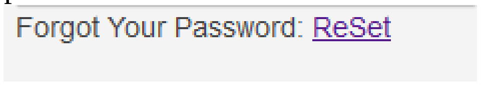
Test number	**Steps to test**	**Expected results**	**Status**
1	Enter correct employee ID Enter correct password Click on sign-in	Successfully login to the system  Signed in as:E2 User Type: System User Sign Out	Pass
2	Enter correct employee ID Enter incorrect password Click on sign-in	Display pop up message Invalid User name or password  Invalid User Name Or Password OK	Pass
3	Enter incorrect employee ID Enter correct password Click on sign-in		Pass
4	Enter incorrect employee ID Enter incorrect password Click on sign-in		Pass
5	Click on reset password Enter email Address submit	Receive a code through email to reset password  Forgot Your Password: ReSet	Pass

Table E.1: Test result for user authentication

Test results for add employee

The test result for the Add employee are given in table E.2

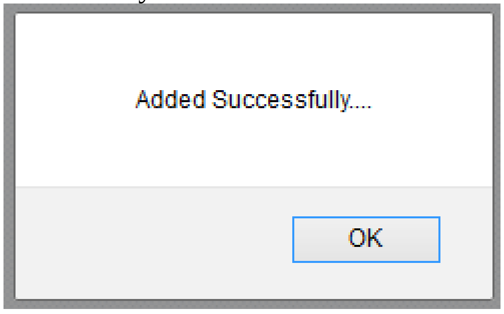
Test number	Steps to test	Expected results	Status
1	Add relevant details for each field in the form	Add the details to the employee table as a new row	Pass
2	Try to add details with empty fields	Display error messages relevant to each field.	Pass
3	Enter incorrect name	Display error message Invalid Name Invalid Name	Pass
4	Enter incorrect NIC	Display error message Invalid NIC Invalid NIC	Pass
5	Enter incorrect email	Display error message Invalid Email Invalid Email	Pass
6	Enter incorrect telephone number	Display error message Invalid Telephone number Invalid Telephone Number	Pass
7	Enter incorrect salary	Display error message Salary should be numeric Salary Should Be Numeric	Pass
8	Enter different passwords	Display error message Password do not match Passwords do not match	Pass
9	Did not select birth day	Display error message Invalid birth day Invalid Birthday	Pass
10	Did not select recruitment day	Display error message Invalid recruitment day Invalid Recruitment Date	Pass
11	Did not select gender	Display error message select gender Select Gender	Pass
12	Click on submit button to submit added details	Display pop up message Added Successfully  Added Successfully.... OK	Pass

Table E.2: Test result for add employee

Test results for add leaves

The test result for the Add leaves are given in table E.3

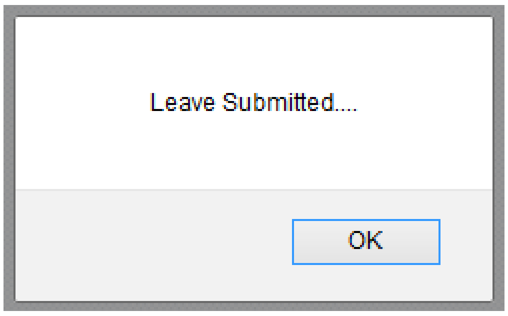
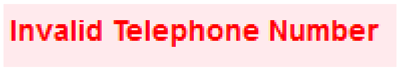
Test number	Steps to test	Expected results	Status
1	Add relevant details for each field in the form	Add the details to the leaves table as a new row	Pass
2	Try to add details with empty fields	Display error messages relevant to each field.	Pass
3	Click on submit button to submit added details	Display pop up message request send for approval  Leave Submitted....	Pass

Table E.3: Test result for Add leaves

Test result for add customer

The test result for the Add customer are given in table E.4

Test number	Steps to test	Expected results	Status
1	Add relevant details for each field in the form	Add the details to the customer table as a new row	Pass
2	Try to add details with empty fields	Display error messages relevant to each field.	Pass
3	Enter incorrect name	Display error message Invalid Name Invalid Name	Pass
4	Enter incorrect email	Display error message Invalid Email Invalid Email	Pass
5	Enter incorrect telephone number	Display error message Invalid Telephone number  Invalid Telephone Number	Pass
6	Click on submit button to submit added details	Display pop up message Added Successfully	Pass

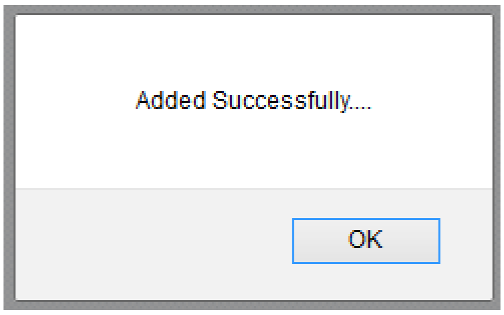
		 Added Successfully.... OK	

Table E.4: Test result for add customer

Test result for search room booking

The test result for the search room booking are given in table E.5

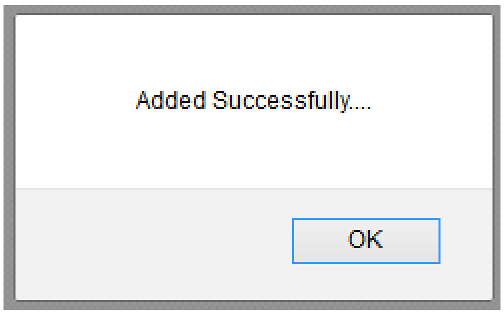
Test number	Steps to test	Expected results	Status
1	Add relevant details for each field in the form	Add the details to the booking table as a new row	Pass
2	Try to add details with empty fields	Display error messages relevant to each field.	Pass
3	Add check in date after check out date	Display error message Checkout cannot be before check in **Checkout cannot be before Checkin**	Pass
4	Click on submit button to submit added details	Display pop up message Added Successfully  Added Successfully.... OK	Pass

Table E.5: Test result for search room booking

Test result for payment

The test result for payment are given in table E.6

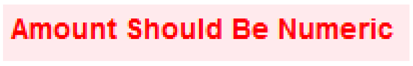
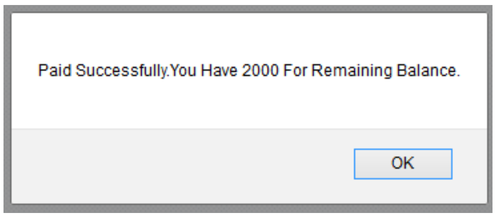
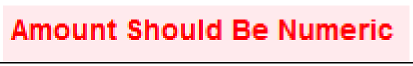
Test number	Steps to test	Expected results	Status
1	Add relevant details for each field in the form	Add the details to the payment table as a new row	Pass
2	Try to add payment amount with empty field	Display error messages amount should be numeric	Pass
3	Enter incorrect payment	Display error message Payment is incorrect  Amount Should Be Numeric	Pass
4	Click on submit button to submit added details	Display pop up message Paid Successfully. Receive an email about payments.  Paid Successfully.You Have 2000 For Remaining Balance. OK	Pass

Table E.6: Test result for payment

Test result for add room type/reception hall

The test result for add room type/reception hall are given in table E.7

Test number	Steps to test	Expected results	Status
1	Add relevant details for each field in the form	Add the details to the room type/ reception hall table as a new row	Pass
2	Try to add price amount with empty field	Display error messages amount should be numeric	Pass
3	Enter incorrect price	Display error message amount is incorrect  Amount Should Be Numeric	Pass
4	Click on submit button to submit added details	Display pop up message Added Successfully.	Pass

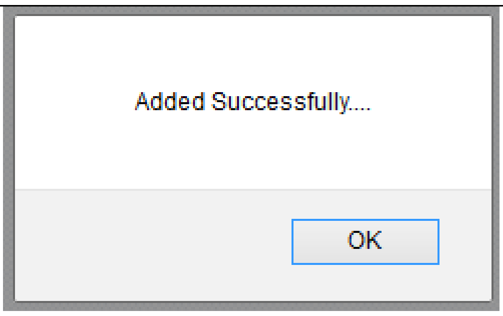
		 Added Successfully....	

Table E.7: Test result for add room type/reception hall

APPENDIX F – CODE LISTING

This appendix is written in order to give the reader a better understanding of the underlying logic and coding methods used in this project. The important code segments which are not included in implementation chapter are included in here. Please refer the supplementary CD for complete code.

Report Generation

These methods generate reports appropriately to the user and save them in PDF format.

```
  <script type="text/javascript">
        function getPdf() {
            var pdf = new jsPDF('p', 'pt', 'letter');

            source = $('#tableData')[0];

            specialElementHandlers = {
                '#bypassme': function(element, renderer) {
                    return true
                }
            };
            margins = {
                top: 80,
                bottom: 60,
                left: 40,
                width: 522
            };
            pdf.fromHTML(
                    source, // HTML string or DOM elem ref.
                    margins.left, // x coord
                    margins.top, {// y coord
                  'width': margins.width, // max width of content
on PDF
                            'elementHandlers': specialElementHandlers
                    },
            function(dispose) {
                var date = new
Date().toDateString().concat(".PDF");
                var title="Booking_";
                var savename=title.concat(date);
                pdf.save(savename);
            }
            , margins);
        }
    </script>
```

Security features

- **Sign in controller**

```
if (isset($_POST["submit"])) {
     $conn = Connection::getInstance();
     $type = $_POST["type"];
     $username = $_POST["username"];
     $password = $_POST["password"];
     if (strcmp($type, '1') == 0) {
          $sql = "SELECT * from systemuser where
employeeId='$username' and userPassword='$password'";
          $result = mysqli_query($conn, $sql);
          $rowcount = mysqli_num_rows($result);
          if ($rowcount > 0) {

header("Location:http://localhost/Hotel_Management_System/
Views/SystemUserHome.php?username=$username&type=System
User");
               exit;
          } else {
               echo '<script>alert("Invalid User Name Or
Password");</script>';
          }
     } elseif (strcmp($type, '2') == 0) {
          $sql = "SELECT * from employee where
employeeId='$username' and password='$password'";
          $result = mysqli_query($conn, $sql);
          $rowcount = mysqli_num_rows($result);
          if ($rowcount > 0) {

header("Location:http://localhost/Hotel_Management_System/
Views/EmployeeHome.php?username=$username&type=Employee");
               exit;
          } else {
               echo '<script>alert("Invalid User Name Or
Password");</script>';
          }
     }
}
```

- **Security of the system handles through priority levels.**

```
    $conn = Connection::getInstance();
    $username = $_GET["username"];
    $type = $_GET["type"];
    $url=$_GET['url'];
    $sql = "SELECT * from systemuser where
employeeId='$username'";
    $result = mysqli_query($conn, $sql);
    while ($row = mysqli_fetch_array($result)) {
            $priority=$row['priority'];}
    if(strcmp($priority,'2')==0 || strcmp($priority,'3')==0){

header("Location:http://localhost/Hotel_Management_System/Vie
ws/".$url."?username=$username&type=System User");
            exit;
    }
    else{
         echo '<script>alert("Sorry. Access Is
Denied.");</script>';
    }
```

Auto Generate Employee IDs when adding employee

```
<?php

    include ("..//Common/Connection.php");
    $conn = Connection::getInstance();
    $query ='SELECT * FROM employee' ;
    $result= mysqli_query($conn,$query);
    $num=  mysqli_num_rows($result);
    $num=$num+1;
    $id='E'.$num;
    ?>
```

```
include ("..//Common/Connection.php");
    $id = $_POST["id"];

    $conn = Connection::getInstance();
    $query ="SELECT * FROM employee where employeeId='$id'" ;
    $result= mysqli_query($conn,$query);
    $row = mysqli_fetch_row($result);
    unset($row[17]);

    $query=mysqli_query($conn,"SELECT * FROM employee where
employeeId='$id'");
    while($row1=mysqli_fetch_array($query))
   {
    $row[]=base64_encode($row1['file']);
   }
```

Validation

The user input for the login is validated through a JavaScript function before directing it to the servlet.

- **Validate NIC**

```
if (nic.value == '' || !NICValidate(nic.value)) {
        nic.focus();
        document.getElementById('nic_err').innerHTML =
"Invalid NIC";
        document.getElementById('nic_err').style.color =
"red";
        validate = false;
    }

function NICValidate(nic) {

    if (nic.length == 10) {
        if (nic.slice(-1) == 'V' || nic.slice(-1) == 'v'
|| nic.slice(-1) == 'X' || nic.slice(-1) == 'x') {
            if (parseInt(nic.slice(0, -1))) {
                return true;
            }
            else {
                return false;
            }
        }
        else {
            return false;
        }
    }
    else if (nic.length == 12) {
        if (parseInt(nic)) {
            return true;
        }
        else {
            return false;
        }
    }
    else {
        return false;
    }
}
```

- **Validate Password**

```
function checkPassword(password,confirmpassword,validate)
  {
    if(password.value != "" && password.value ==
confirmpassword.value) {
      if(password.value.length < 6) {

document.getElementById('password_err').innerHTML="Passwo
rd must contain at least six characters";

document.getElementById('password_err').style.color="red"
;
        validate=false;
      }
      re = /[0-9]/;
      if(!re.test(password.value)) {

document.getElementById('password_err').innerHTML="passwo
rd must contain at least one number (0-9)";

document.getElementById('password_err').style.color="red"
;
        validate=false;
      }
      re = /[a-zA-Z]/;
      if(!re.test(password.value)) {

document.getElementById('password_err').innerHTML="passwo
rd must contain at least one  letter ";

document.getElementById('password_err').style.color="red"
;
        validate=false;
      }

    } else {

document.getElementById('password_err').innerHTML="Passwo
rds do not match";

document.getElementById('password_err').style.color="red"
;
      validate=false;
    }

  }
```

- **Validate Email**

```
function validateEmail(email) {
    var re =
/^(([^<>()\[\]\\.,;:\s@"]+(\.[^<>()\[\]\\.,;:\s@"]+)*)|(".
+"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-
9]{1,3}])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/;
    return re.test(email);
}
```

- **Validate Date**

When making a booking check the checking date with the current date. After that check check-in and check-out validity.

```
function checkDate(datestr) {
        var now = new Date();
        var dt1 = Date.parse(now);
        dt2 = Date.parse(datestr);
        if (dt2 < dt1) {
            return false;
        }
        else{
        return true;
    }
 }

 function compareDates(checkin,checkout) {
        var dt1 = Date.parse(checkin);
        dt2 = Date.parse(checkout);
        if (dt2 < dt1) {
            return false;
        }
        else{
        return true;
    }
 }
```

Insert Data

```
 $sql="INSERT INTO employee
(employeeid,nic,name,address,userName,password,gender,sal
ary,position,birthDay,recruitmentDate,noOfCasualLeaves,no
OfMedicalLeaves,telephone,email,currentCasualLeaves,curre
ntMedicalLeaves,file)". " VALUES ('" . $employeeid. "','"
. $nic. "','" . $name. "','" . $address. "','" .
$username. "','" . $password. "','" . $gender. "','" .
$salary. "','" . $position. "','" . $birthday. "','" .
$recruitmentdate. "','" . $casualleaves. "','" .
$medicalleaves. "','" . $telephone. "','" . $email. "','"
. $casualleaves. "','" . $medicalleaves. "','" . $image.
"')";
 $result= mysqli_query($conn,$sql);
 if($result) {
 echo '<script>alert("Added Successfully");</script>';
```

Update Data

```
$sql="UPDATE employee SET
nic='$nic',name='$name',address='$address',userName='$users
name',gender='$gender',salary='$salary',position='$position
',birthDay='$birthday',recruitmentDate='$recruitmentdate',n
oOfCasualLeaves='$casualleaves',noOfMedicalLeaves='$medical
leaves',telephone='$telephone',email='$email',file='$image'
WHERE employeeid='$employeeid'";
    $result= mysqli_query($conn,$sql);
    if($result) {
    echo '<script>alert("Updated Successfully");</script>';
```

Database connection

Database connection uses a common class with the common implementation. There is a static function which returns a connection. Also there are getters and setters to set parameters.

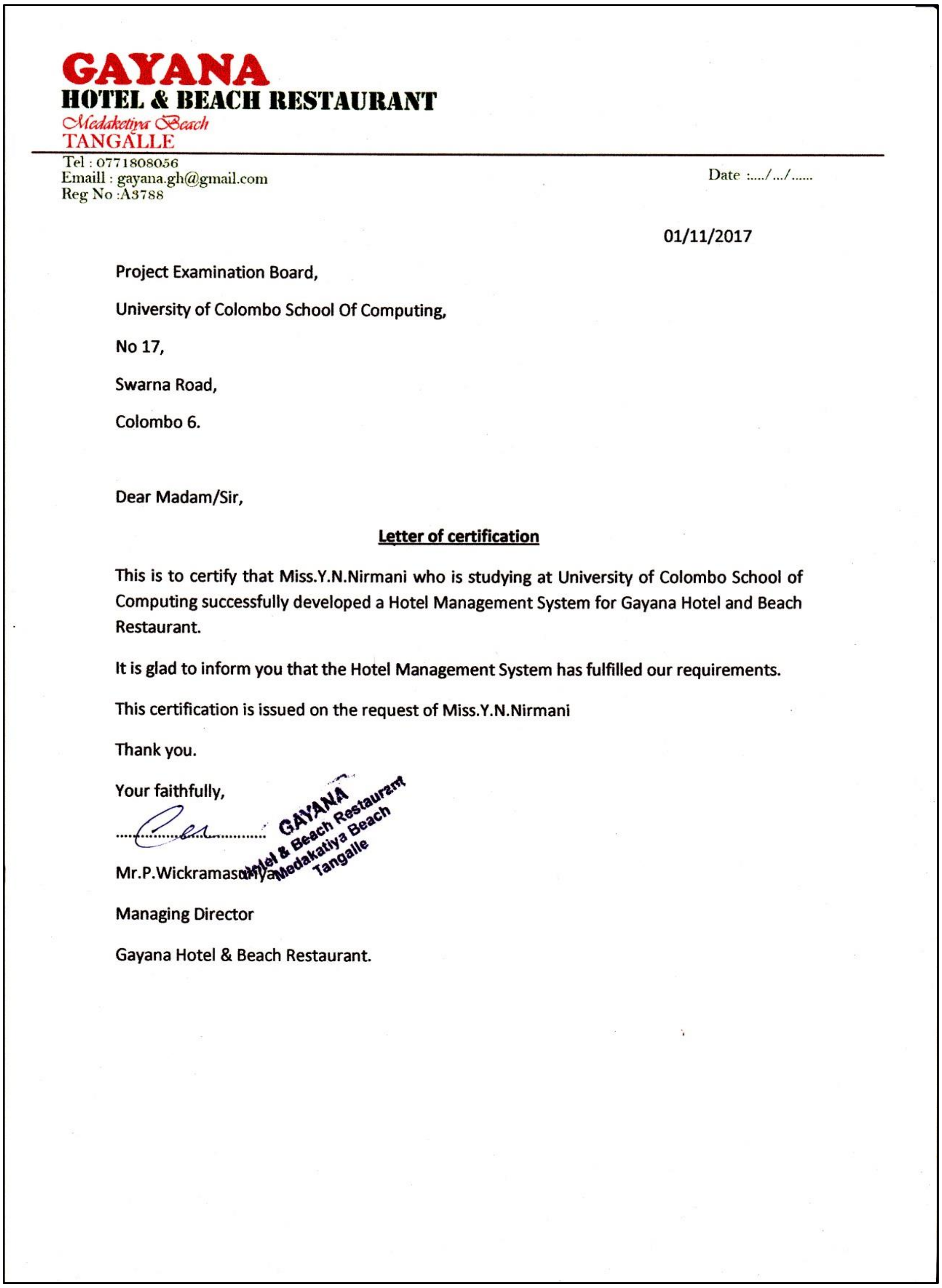
```
public static function getInstance() {
        $conn =
mysqli_connect("localhost","root","","hotelmanagementdb
");
        return $conn;
    }
    private function __construct() {
        $this->_connection = new mysqli($this->_host,
$this->_username,
            $this->_password, $this->_database);
        if(mysqli_connect_error()) {
            trigger_error("Failed to conencto to MySQL:
" . mysql_connect_error(),
                E_USER_ERROR);
        }
    }
```

Update Approve Leave Data

```
$id = $_POST["id"];
$status = $_POST["status"];
$date = $_POST["date"];
$conn = Connection::getInstance();
$query = "SELECT * from leaves where id='$id'";
$result = mysqli_query($conn, $query);
while ($row = mysqli_fetch_array($result)) {
    $employeeid = $row['employeeid'];
    $type = $row['type'];
}
$query = "UPDATE leaves SET status='$status' where
id='$id'";
$result = mysqli_query($conn, $query);
$query = "SELECT * from employee where
employeeId='$employeeid'";
$result = mysqli_query($conn, $query);
while ($row = mysqli_fetch_array($result)) {
    $casualleaves = $row['currentCasualLeaves'];
    $medicalleaves = $row['currentMedicalLeaves'];
```

```
$subject = "Leave Request";
if (strcmp($status, "1") == 0) {
    if (strcmp($type, "1") == 0) {
        $casualleaves = $casualleaves - 1;
    } else {
        $medicalleaves = $medicalleaves - 1;
    }
    $query = "UPDATE employee SET
currentCasualLeaves='$casualleaves',currentMedicalLeaves='$m
edicalleaves' WHERE employeeId='$employeeid'";
    $result = mysqli_query($conn, $query);
    $message = "Your Leave Request On " . $date . " Has Been
Approved.";
    EmailController::sendEmail($email, $message, $subject);
} else {
    $message = "Your Leave Request On " . $date . " Has Been
Rejected.";
    EmailController::sendEmail($email, $message, $subject);
}
```

APPENDIX G – CLIENT CERTIFICATE



GAYANA
HOTEL & BEACH RESTAURANT
Medaketiya Beach
TANGALLE

Tel : 0771808056
Emaill : gayana.gh@gmail.com
Reg No :A3788

Date :...../.../......

01/11/2017

Project Examination Board,

University of Colombo School Of Computing,

No 17,

Swarna Road,

Colombo 6.

Dear Madam/Sir,

Letter of certification

This is to certify that Miss.Y.N.Nirmani who is studying at University of Colombo School of Computing successfully developed a Hotel Management System for Gayana Hotel and Beach Restaurant.

It is glad to inform you that the Hotel Management System has fulfilled our requirements.

This certification is issued on the request of Miss.Y.N.Nirmani

Thank you.

Your faithfully,

[illegible]

GAYANA
Hotel & Beach Restaurant
Medakatiya Beach
Tangalle

Mr.P.Wickramasuriya

Managing Director

Gayana Hotel & Beach Restaurant.

GLOSSERY

Apache	- Secure web server developed by apache software foundation
Black box testing	- Testing technique use to test functionalities of an application
Class diagram	-A class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system.
CSS	-Style sheet language used for describing a presentation of a document written in markup language
Graphical User Interface	- is a type of user interface that allows users to interact with electronic devices with images rather than text commands.
JavaScript	- is a prototype-based, object-oriented client-side scripting language that is dynamic. It is also considered as a functional programming language.
PHP Hyper-text Pre-processor (PHP**)**	- is a server-side programming language.
Sequence diagram	- A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order.
SQL	-is a special-purpose programming language designed for managing data held in a relational database management system

Use case diagram	-A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved
White box testing	-Testing technique used specifically on testing internal knowledge and the structure of the software.
XAMPP	- Open source Bundled software package. Include Apache, PHP, MYSQL, and Perl.

INDEX