HOTEL MANAGEMENT SYSTEM FOR HOTEL SUMADAI.

L.L.D.C.CHATHURANGI

Registration No:R**110198**

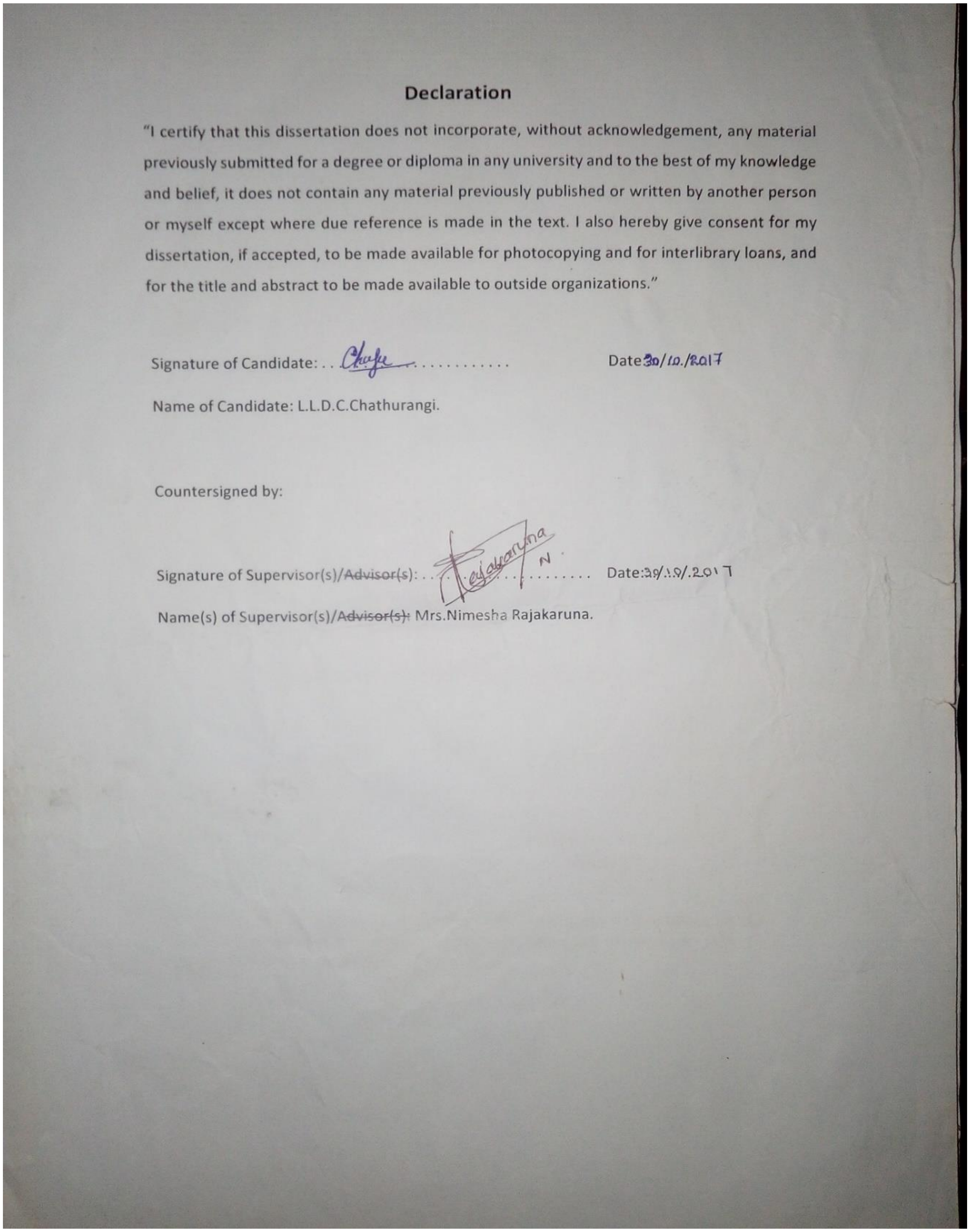
Index No: **1101986**

Supervisor Name:Mrs.N.Rajakaruna

December 2017

This dissertation is submitted in partial fulfilment of the requirement of the Degree of Bachelor of Information Technology (external) of the University Of Colombo School Of Computing

DECLARATION



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: . . . . Date 30/10./2017

Name of Candidate: L.L.D.C.Chathurangi.

Countersigned by:

Signature of Supervisor(s)/~~Advisor(s)~~: . . Rajakaruna N. Date:30/10/.2017

Name(s) of Supervisor(s)/~~Advisor(s):~~ Mrs.Nimesha Rajakaruna.

ABSTRACT

Hotel Sumadai is a famous and beautiful hotel in Beruwala. It has 21 guest room of various types and having an Ayurveda spa. This project aims at creating hotel management system which can be used by customers to reserve hotel rooms. Users can register and log into the system. The administrator can know the details and reservation and daily income.

To overcome these problems Sumadai has decided to use Information Technology with their all department activities. This Hotel Management System will cover, online reservation system, employee information management, customer information management, stock management, room management, maintenance management, Ayurveda spa management, payroll management and login functionalities.

Hotel Sumadai currently using manual system which takes more time and therefor it is not that effective. They are interested in introducing a computerized system to handle the database. XAMMP server is used in development with apache as the web server. By system in order to increase the efficiency of the services.

It was decided to use object-oriented techniques for analysis, design and development of the system. UML diagrams have been used to analyses requirements and draw the design. The architecture used for the system is MVC architecture. The system has to be hosted on a server and the staff related to project activities must be able to access the system. PHP server side scripting language is used to develop the system and MySQL is used to develop system .users will able to overcome the problems of the current manual system and will be able to perform their tasks efficiently and effectively.

ACKNOWLEGEMENT

I would like to express my sincere thanks to my supervisor Mrs.Nimesha Rajakaruna from Esoft Metro Campus for all her ideas and suggestions and especially for the guidance throughout the difficulties encountered during the course of the Hotel Management project. Further special thanks to Mr.Sunil Mendis, Owner of Hotel Sumadai for giving the opportunity to do a project in the hotel, and I would like to thank all the hotel staff those who gave me an unending support throughout the whole project with providing necessary information and guidance. Finally, I would also like to thank to my dearest parents, brother and friends who were with me every time and encouraged me to make this goal success.

Table of Contents

List of Figures

List of Tables

CHAPTER1
INTRODUCTION

1. INTRODUCTION

This chapter basically covers motivation for the project, what the project is about, its nature and the scope, objectives of the project, problem to be solved and the proposed solution of the project.

1.1 MORTIVATION OF THE PROJECT

Hotel Sumadai is a famous and beautiful hotel in Beruwala. They owns 21 guest room of various types and having Ayurveda spa.They have good name in that area.

Hotel Sumadai currently using manual system which takes more time and there for it is not that effective. They are interested in introducing a computerized system in order to increase the efficiency of the services.

Hotel Sumadai is interested in introducing a Hotel Management System in order to automate their activities. Being interested in web application development I see this is as an opportunity to apply my knowledge and theories that I gained during the BIT degree program to a practical situation.

1.2 OBJECTIVE OF THE PROJECT

- Converting the current ineffective, time consuming manual system at the Hotel Sumadai in to a much easier and effective automated system.
- Online reservation is an effective way of booking rooms in the modernized busy world.
- This will save consumer’s valuable time.
- Generate reports and let the decision makers to take decisions in an effective manner.
- Handle customer orders effectively in order to meet the requirements of the consumers and increase customer relationship through effective communication.

1.3 SCOPE OF THE PROJECT

-Online Reservation System.

Online reservation system which can be used by customers to reserve hotel room. Users can register and log in to the system.

-Employee information management.

Manage employee personal details, allowances and leave details.

-Customer information management.

Manage company's interaction with customers. Keeping customers updated with the latest promotional packages, and sending appreciation emails are just few clicks away.

-Stock management system.

System should be able add, edit, view and delete stock details and system should be colouring the rows when the number of stocks are increased and decreased. Calculated available stock.

-Manage room inventory.

Administrator can know which guests have stayed in each one of hotel rooms, and get to know what the available rooms are. In addition, to know how this will benefit hotels' housekeeping and maintenance operations.

-Message alert notification.

Give the message for reservation manager to about booking details.

-Maintenance management.

Manage maintenance detail of hotel rooms and pool.

-Ayurveda spa management.

Manage customer details of customers, rooms, doctors and medicines

-Payroll Management.

Involves everything that has to do with the payment of employees and the filing of employment taxes. This includes keeping track of hours, calculating wages withholding taxes and other deductions.

1.4 HARDWARE AND SOFTWARE REQUIREMENTS

	HARDWARE	**SOFTWARE**
SERVER	Intel Pentium Core 3 processor 500 GB Hard Disc with 250 GB free space	Microsoft Windows 10 XAMPP Server-Version 2.5 Apache - 2.2.21 PHP - 5.3.8 MySql - 5.5.16 Net Beans IDE 7.0 / Dreamweaver Rational Rose 7 Adobe Photoshop
CLIENTS	Intel Pentium Dual Core processor 1 GB RAM 80 GB Hard Disc with 10 GB free space	Microsoft Windows XP and above Any Web Browser

Table1.4 Hardware Software Requirements.

1.5 STRUCTURE OF THE DISSERTATION

The dissertation provides an overall idea about the development process of the Operation Management System. Dissertation is divided into 6 main chapters.

CHAPTER 2 – ANALYSIS

The analysis chapter explains the fact finding techniques used, the current manual system and the functional and non-functional requirements of the proposed system. Top level use case diagram also included for a better understating of the system.

CHAPTER 3 – DESIGN

Design chapter explains the software process models, process modal used for development with justification and the design architecture of the system. It also includes use case diagrams, class diagrams and use case narratives. Module design, User interface design and database design also explained here.

CHAPTER 4 – IMPLEMENTATION

Implementation chapter describes the hardware and software environment used while development. It also lists out reused components and major code segments. The network architecture and the application architecture of the system also described here.

CHAPTER 5 – EVALUATION

Evaluation chapter explains the testing methods used and how the system is tested along with test cases and results.

CHAPTER 6 – CONCLUSION

Last chapter explains critical evaluation, future enhancements and lessons learnt during the project.

Appendixes

These are provided further details about the content of the dissertation chapters which were not included in the chapters.

CHAPTER 2
ANALYSIS

2. ANALYSIS

2.1 INTRODUCTION

System analysis is an important phase in system development. In this phase the system requirements were collected and understood with the help of the users. User involvement is a key factor in this phase and appropriate techniques were used to gather the requirements.

2.2 FACT FINDING TECHNIQUES

Understanding the system requirements clearly has a major effect on final outcome of the project. Several fact finding techniques were used in system analysis. Sampling of existing Documentation is used as the primary fact finding technique since there is a current manual system. Forms, Reports and various kinds of documents were used to identify the requirements of the system.

Interviewing was also used as a fact finding technique and it was helpful to understand the current manual system along with sampling of existing documentation. Interviewing was carried out in a more formal and structured way with the divisional heads. Interviewing the employers was very important since they are the main users of the proposed system.

Observation technique was also used in fact finding and it was helpful to identify the flow of certain documents which are going to be important to the system. But certain employees were not feeling comfortable with this technique. Some additional requirements were proposed by the employers which were not part of the current manual system. Some of those additional requirements were approved by the divisional heads.

2.3 PROBLES WITH REQUIREMENT GATHERING.

The system has to be developed based on the information gathered from the client and the system users. The client was not a regular IT user and because of that collecting technical information for the system was a big hassle. Information gathered from users like managers supervisors and room boys of the hotel who were also not fluent in Technical IT related information.

Prototypes of the system had to be used to explain and gather the wanted functionalities and to clarify whether they are feasible.

2.4 ANALISING THE CURRENT MENUAL SYSTEM.

Online Reservation System.

When the customer gives their information, they store reservation detail manually. Because of that they have waste Lot of time, store these details. This system provide Availability check and instant confirmations eliminate email exchanges between guests and hotel reservations department. Valuable reservations staff time is saved because reservation requests will no longer have to be processed. And Customer can log, register and reserve hotel rooms. Can check availability of rooms. Also customer can do online payment via system.

Employee information management.

Manual handling of employee information poses a number of challenges. This is evident in procedures such as leave management where an employee is required to fill in a form which may take several weeks or months to be approved. The use of paper work in handling some of these processes could lead to human error, papers may end up in the wrong hands and not for getting the fact that this is time consuming. Using new system, they can do there works so easily and saved valuable time.

Customer information management.

The collected information from customer included for example: email address, phone number and address information. There were several issues with this manual approach. The data collected varied from employee to another. The handwriting was unclear when it was typed in a hurry. Searching for a certain customer was tedious. Using new system, they can do their work so easily.

Stock management system.

There were several issues with this manual approach. The data collected varied all stock to another. The handwriting was unclear when it was written in a hurry. Searching for a stock details was tedious. Have to calculate available stock detail manually. Using new system, they can add, edit, update data easily and auto calculate auto metrically.

Room management.

According to manual system they have to write all the details of rooms. Searching for available room was difficult. Using new system remembers which guests have stayed in each one of hotel rooms. And get idea what are the available rooms very easy.

Maintenance management.

There were several issues with this manual approach. The handwriting was unclear when it was written in a hurry. Searching for a certain maintenance details in room was tedious. Using new system, they can do their work so easily.

Payroll Management.

Monthly payments of the all employees are calculated manually. Its waste valuable time. Proposed system has given a solution to store details for each and every employee and manage details efficiently and calculated employee salary without any problem.

2.5 FUNCTIONAL REQUIREMENTS

2.5.1 Online Reservation System.

- Customer can log, register and reserve hotel rooms.
- Can check availability of rooms.
- Can do online payment.
- System has provided facility to generate a report of reserved room details.

2.5.2 Employee information management.

- The system should be able add employee details.
- Able to display all details of customer.
- User can search employee details.
- Able to get leave form of employee.

2.5.3 Customer information management.

- Manage company’s interaction with customers.
- Add, edit and update customer data.
- Keeping customers updated with the latest promotional packages.
- Sending appreciation emails are just few clicks away.

2.5.4 Stock management system.

- System should be able add, edit, view and delete stock details.
- System should be colouring inventory, the rows when the number of stocks are increased and decreased.
- Calculated available stock.

2.5.5 Room management

- Administrator can know which guests have stayed in each one of hotel rooms.
- Get to know what the available rooms are.
- Can get details about room status.
- Show the room status in table and categorized each and every rooms.

2.5.6 Message alert notification.

- Give the message for reservation manager to about booking details.

2.5.7 Maintenance management.

- System can add the maintenance details of hotel rooms and pool.
- Display all the maintenance data of rooms and pool.

2.5.8 Ayurveda spa management.

- System should be able to add customers, rooms, doctors and medicines details.
- Give the availability of Ayurveda saps' rooms, doctors and medicines.

2.5.9 Payroll Management.

- System can includes keeping track of hours.
- Calculating wages withholding taxes and other deductions.
- System should be able to Developing pay slip.
- System has provided facility to generate a report of salary details.

2.5.10 Login functionalities.

- User should be able to login the system.

2.6 NON FUNTIONAL REQUIREMENT

2.6.1 Reliability

The proposed system should be reliable. That is should provide correct output belongs to correct inputs. It should not provide redundant data.

Accuracy of the results should be confirmed by the system

2.6.2 User Friendliness

The proposed system should be more user-friendly. It should contain interfaces that are more convenient and easy to handle.

2.6 .3 Performance

All pages should load within 5-60 seconds at any load condition.

2.6 .4 Availability

Information should be available whenever the user need.

2.6.5 Security

The system should guard against unintended and inadvertent correction of information by the user himself.

2.7 HIGH LEVEL USE CASE DIAGRAM FOR HOTEL MANAGEMENT SYSTEM

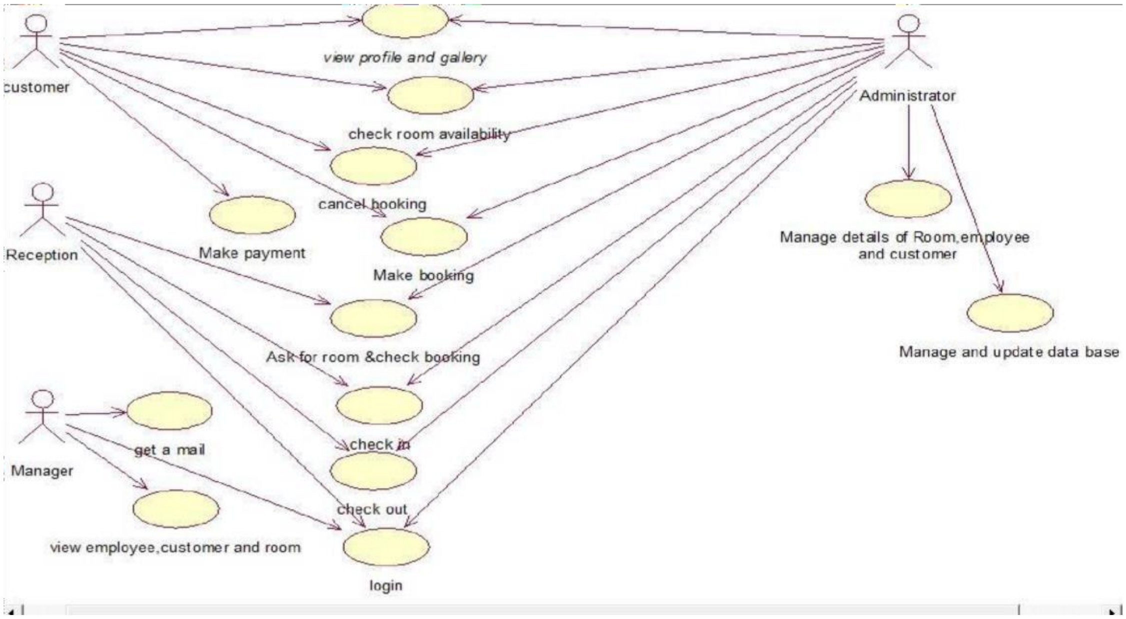
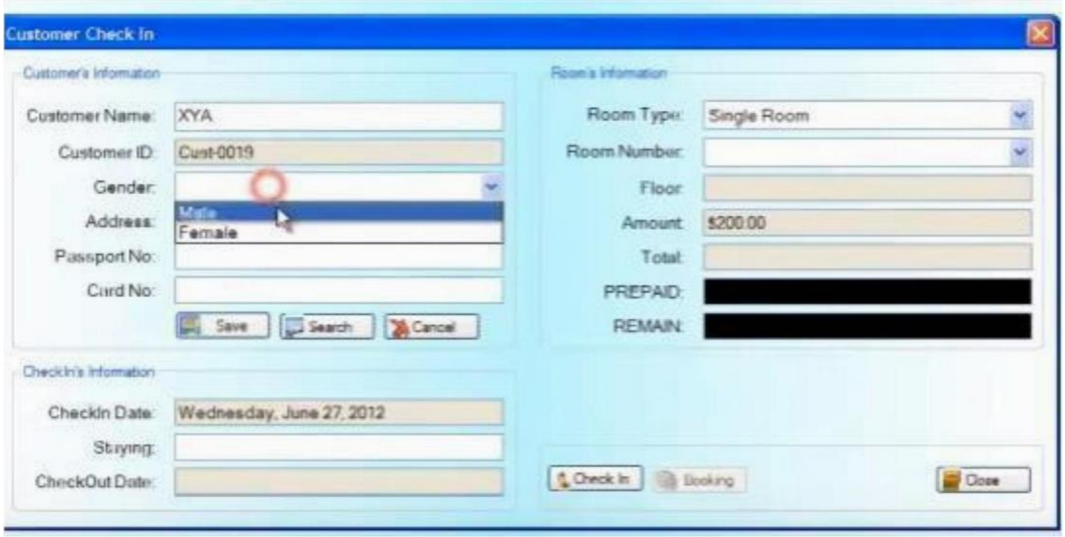


Figure 2. 7.1– Use case Diagram for Hotel management system.

2.7.1 EXISTING FOR SIMILAR SYSTEMS

Literature review and similar system analysis were performed after the requirement gathering stage to redefine the requirements and to have a clear overview of the system. Most of the systems don’t include any data handling or processing systems. They are only including the detail reservation systems. [1] [2] [3][4]



Customer Check In

Customer's Information

Customer Name: XYA

Customer ID: Cust-0019

Gender: Male / Female

Address:

Passport No:

Card No:

Room's Information

Room Type: Single Room

Room Number:

Floor:

Amount: $200.00

Total:

PREPAID:

REMAIN:

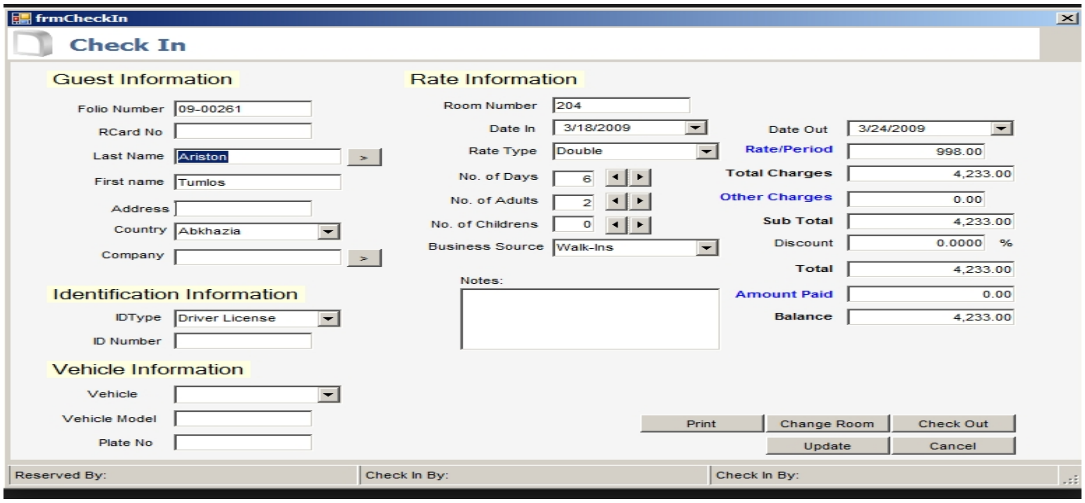
CheckIn's Information

CheckIn Date: Wednesday, June 27, 2012

Staying:

CheckOut Date:

Figure 2.7.1 similar system for hotel management



frmCheckIn

Check In

Guest Information

Folio Number: 09-00261

RCard No:

Last Name: Ariston

First name: Tumlos

Address:

Country: Abkhazia

Company:

Identification Information

IDType: Driver License

ID Number:

Vehicle Information

Vehicle:

Vehicle Model:

Plate No:

Rate Information

Room Number: 204

Date In: 3/18/2009

Rate Type: Double

No. of Days: 6

No. of Adults: 2

No. of Childrens: 0

Business Source: Walk-Ins

Notes:

Date Out: 3/24/2009

Rate/Period: 998.00

Total Charges: 4,233.00

Other Charges: 0.00

Sub Total: 4,233.00

Discount: 0.0000 %

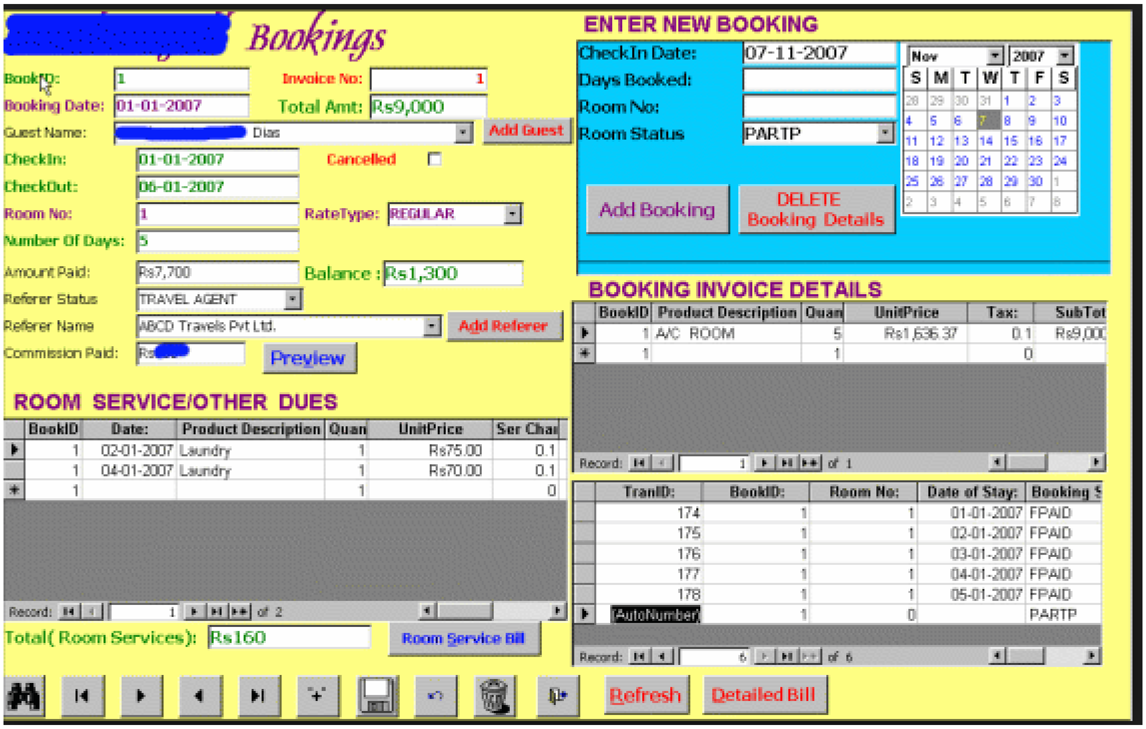
Total: 4,233.00

Amount Paid: 0.00

Balance: 4,233.00

Reserved By: | Check In By: | Check In By:

Figure 2.7.2 similar system for hotel management



[redacted] Bookings

BookID: 1 Invoice No: 1
Booking Date: 01-01-2007 Total Amt: Rs9,000
Guest Name: [redacted] Dias
CheckIn: 01-01-2007 Cancelled ☐
CheckOut: 06-01-2007
Room No: 1 RateType: REGULAR
Number Of Days: 5
Amount Paid: Rs7,700 Balance : Rs1,300
Referer Status TRAVEL AGENT
Referer Name ABCD Travels Pvt Ltd.
Commission Paid: Rs[redacted]

ROOM SERVICE/OTHER DUES

BookID	Date:	Product Description	Quan	UnitPrice	Ser Char
1	02-01-2007	Laundry	1	Rs75.00	0.1
1	04-01-2007	Laundry	1	Rs70.00	0.1
1			1		0

Record: 1 of 2

Total(Room Services): Rs160

ENTER NEW BOOKING

CheckIn Date: 07-11-2007
Days Booked:
Room No:
Room Status PARTP

Nov 2007

S	M	T	W	T	F	S
28	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

BOOKING INVOICE DETAILS

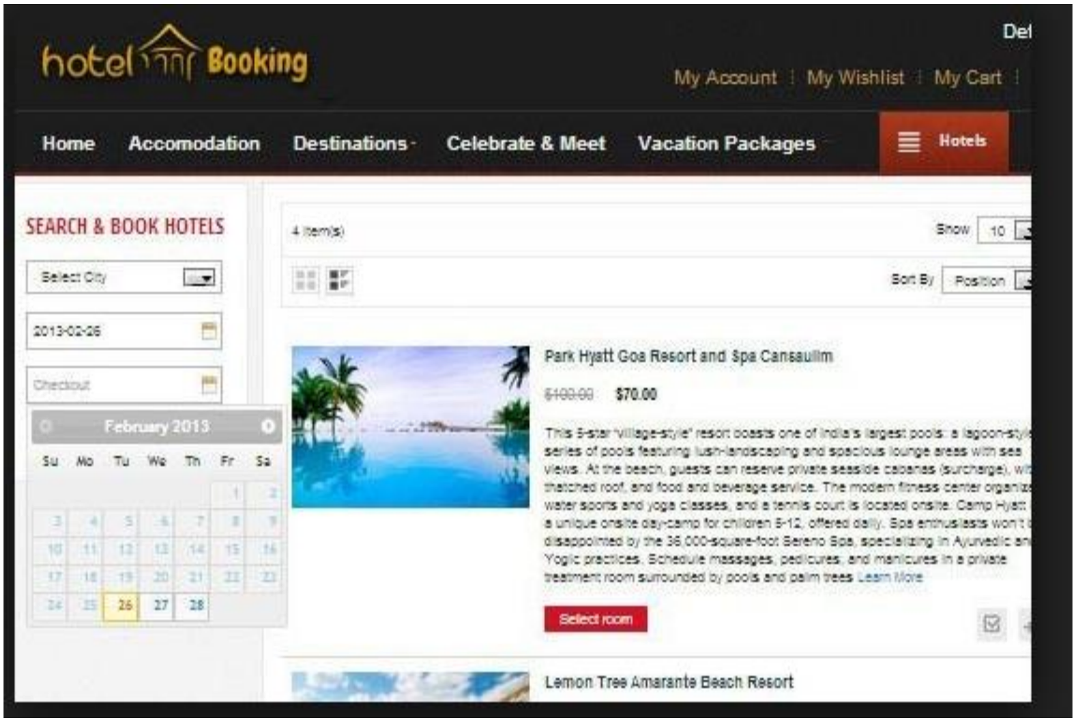
BookID	Product Description	Quan	UnitPrice	Tax:	SubTot
1	A/C ROOM	5	Rs1,636.37	0.1	Rs9,00
1		1		0	

Record: 1 of 1

TranID:	BookID:	Room No:	Date of Stay:	Booking S
174	1	1	01-01-2007	FPAID
175	1	1	02-01-2007	FPAID
176	1	1	03-01-2007	FPAID
177	1	1	04-01-2007	FPAID
178	1	1	05-01-2007	FPAID
(AutoNumber)	1	0		PARTP

Record: 6 of 6

Figure 2.7.3 similar system for hotel management



SEARCH & BOOK HOTELS

Select City
2013-02-26
Checkout

February 2013

Su	Mo	Tu	We	Th	Fr	Sa
					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		

4 item(s) Show 10

Sort By Position

Park Hyatt Goa Resort and Spa Cansaulim

~~$100.00~~ $70.00

This 5-star "village-style" resort boasts one of India's largest pools: a lagoon-styl
series of pools featuring lush-landscaping and spacious lounge areas with sea
views. At the beach, guests can reserve private seaside cabanas (surcharge), wi
thatched roof, and food and beverage service. The modern fitness center organiz
water sports and yoga classes, and a tennis court is located onsite. Camp Hyatt
a unique onsite day-camp for children 5-12, offered daily. Spa enthusiasts won't
disappointed by the 36,000-square-foot Sereno Spa, specializing in Ayurvedic an
Yogic practices. Schedule massages, pedicures, and manicures in a private
treatment room surrounded by pools and palm trees Learn More

Lemon Tree Amarante Beach Resort

Figure 2.7.4 similar system for hotel management

CHAPTER 3
DESIGN

3. DESIGN

3. 1 INTRODUCTION

By reading this chapter reader can get an idea about how the best method to develop the system was selected and what technologies have to be used and what tools are required to develop the system. From information gathered from the Analysis Phase Diagrams such as Use case, Class Diagrams are drawn so support the development of the system with properly defined system boundary’s and functionalities.

3.2 DEVOLOPMENT METHOD.

WATERFALL METHOD

The Waterfall model is a sequential development approach, in which development is seen as flowing steadily downwards (like a waterfall) through the phases of requirements analysis, design, implementation, testing (validation), integration, and maintenance.

[5]

PROTOTYPING

Software prototyping is the development approach of activities during software development, the creation of prototypes, i.e., incomplete versions of the software program being developed. [6]

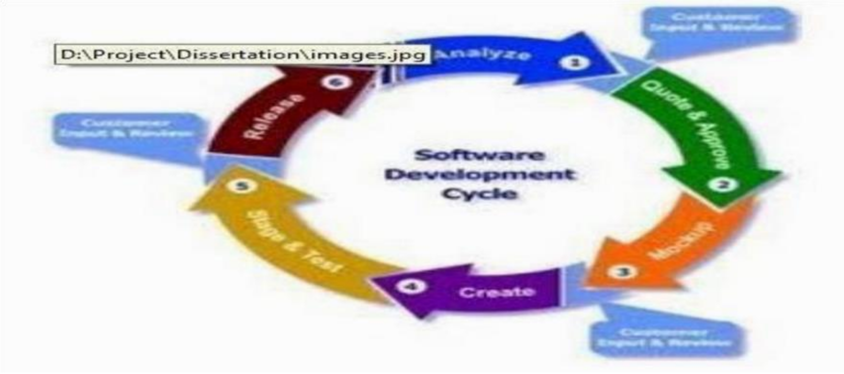


Figure 3.2.1 prototyping model

RATIONAL UNIFIED PROCESS

The Rational Unified Process (RUP) is an iterative software development process framework created by the Rational Software Corporation, a division of IBM since 2003 RUP is not a single concrete prescriptive process, but rather an adaptable process framework, intended to be tailored by the development organizations and software project teams that will select the elements of the process that are appropriate for their needs. RUP is a specific implementation of the Unified Process [7]

3.3 SELECTING A DEVOLOPMENT APPROACH.

Since the project requirements are not well defined and requirements may change along with the development Waterfall Methodology was not chosen as the development approach. In waterfall methodology requirements should be well defined and each phase is started one after other.

Prototyping was highly effective to be used in the project, prototyping is not a methodology it’s a development approach, so it should be used along with a methodology to develop the system. According to the user requirements user requests a low cost effective solution. As the development methodology, hybrid methodology is used which combines both prototyping and iterative development. By using these methods, the system can be developed reliably verifying client requirements.

3.4 HIGH LEVEL USE CASE DIAGRAM FOR SYSTEM

Use case diagrams from the Unified modelling language which is drawn to represent the graphical overview of the system users (actors) and their actions and tasks, and some actors have dependencies which mean they must do before proceeding to the next segment, by drawing the use case diagrams understanding the system functionality and the roles of the system users.

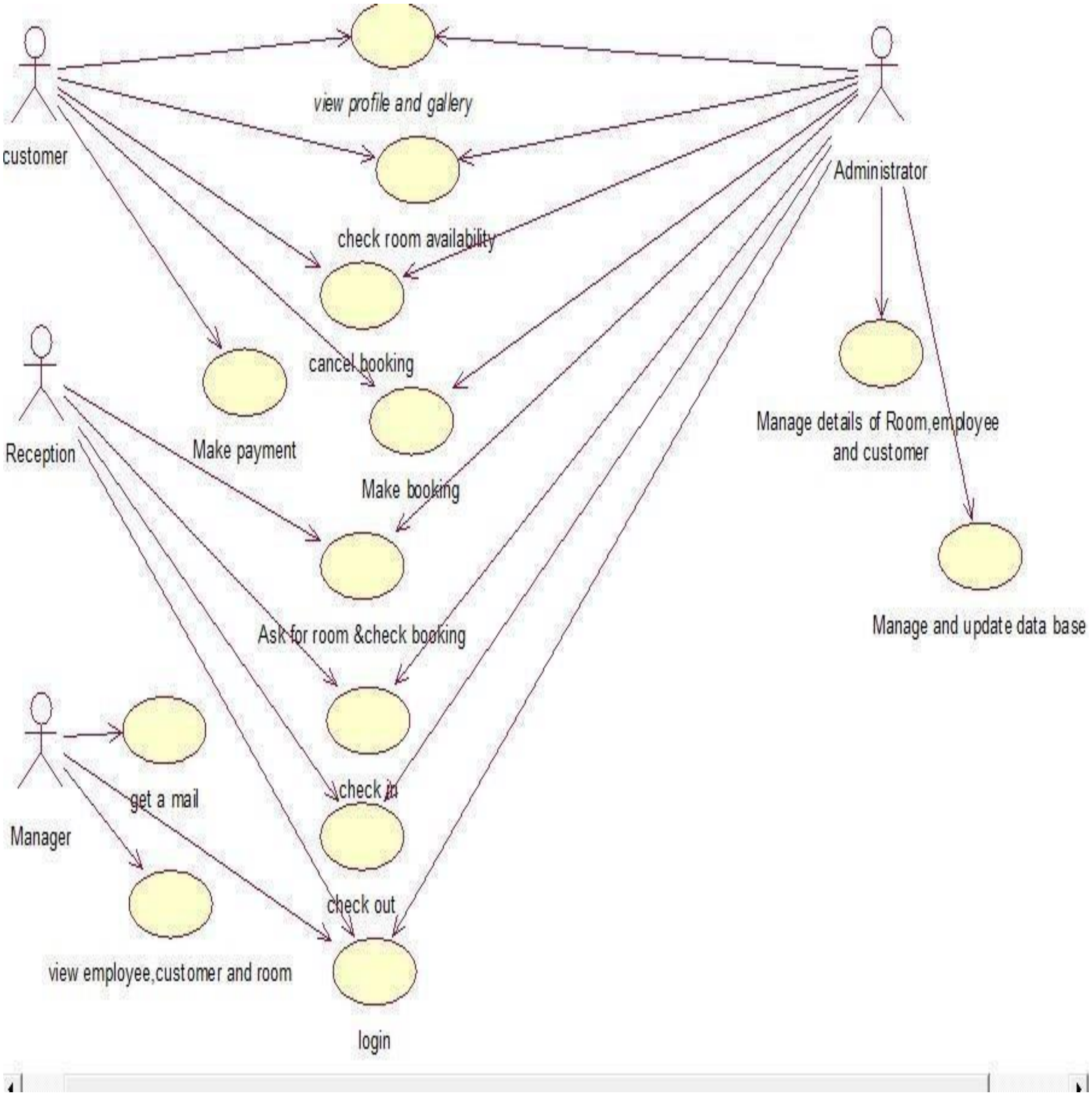


Figure 3.4.1 use case diagram for system.

3.5 Class Diagram for System.

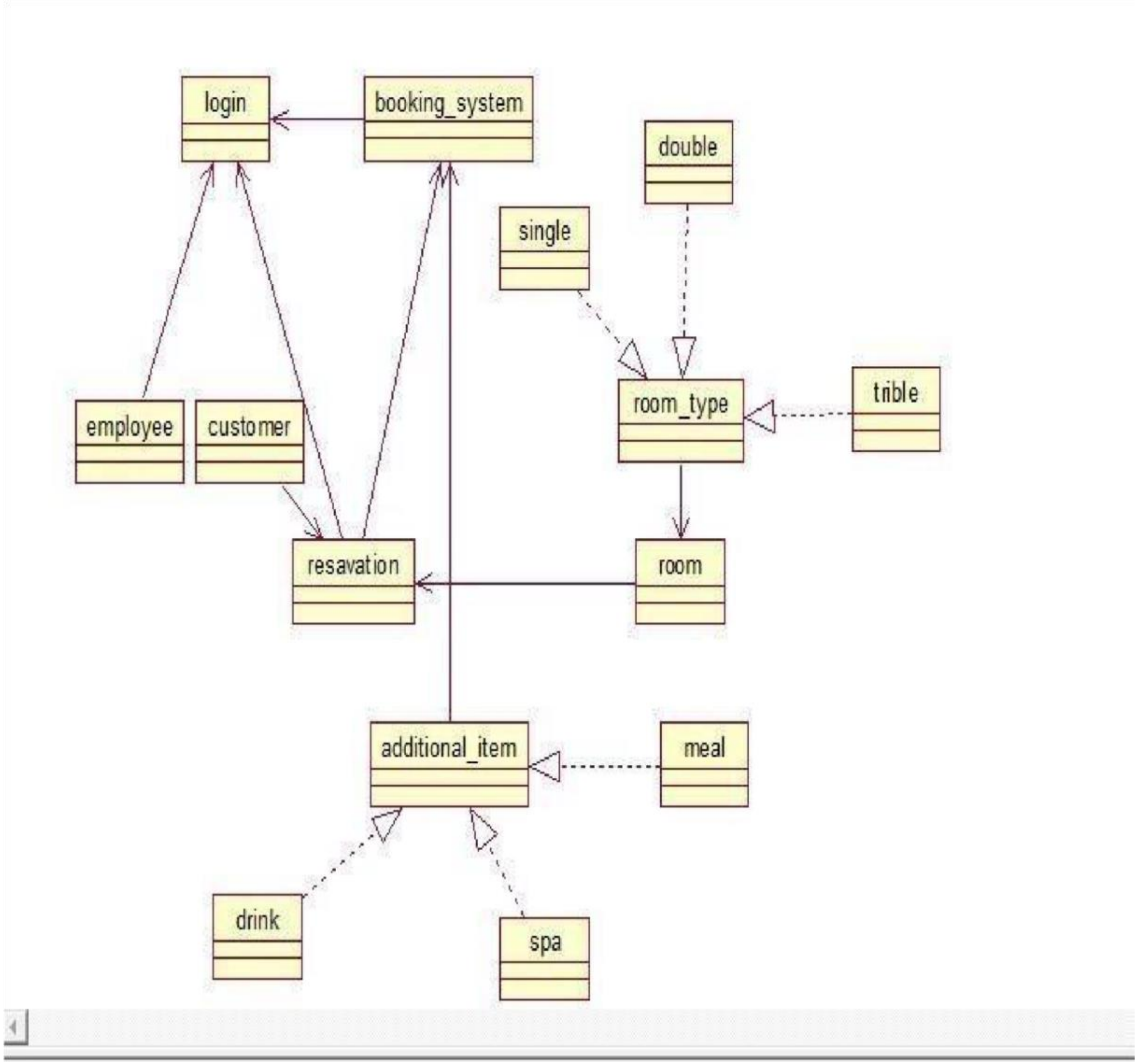


Figure 3.5.1 class diagram for system

3.6 DATABASE DESIGN

Database design is the process of producing a detailed data model of a database.

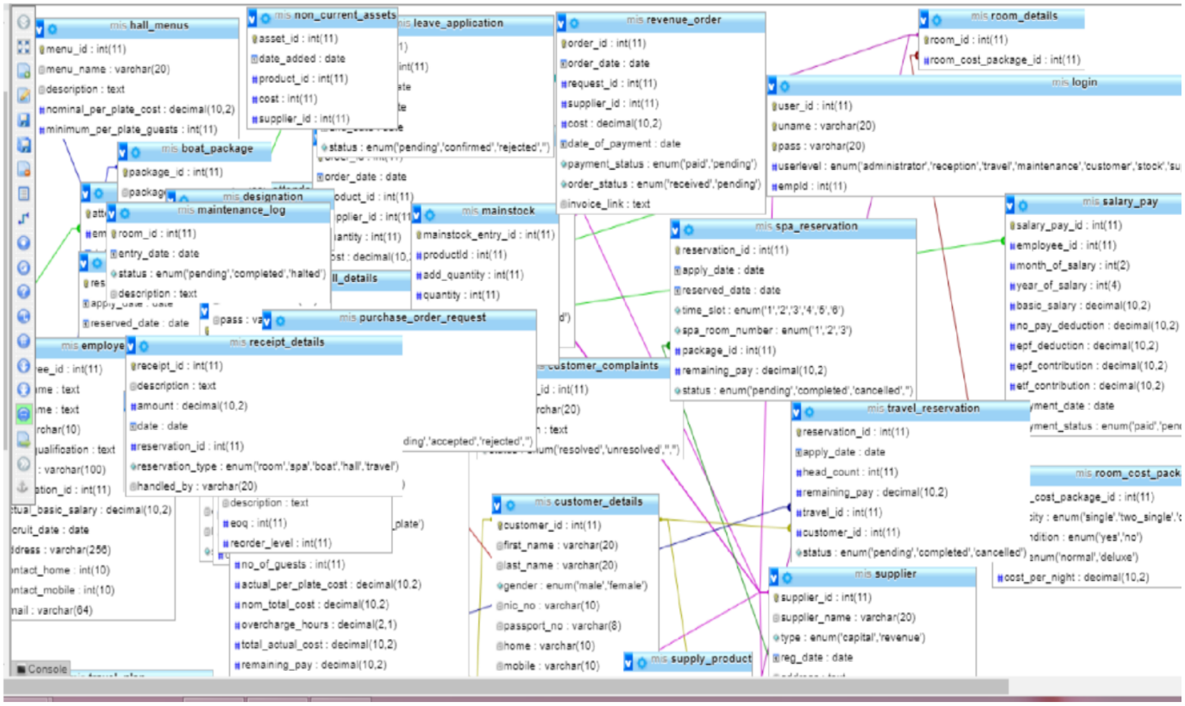


Figure 3.6.1 database design for system

3.7 USER INTERFACE DESIGN

Interface designing part is a critical part of the overall software designing process. Because many user errors can be occurred, if the interface designing is poor. Good interface designing should be match with the user expectation. That is the reason of criticalness of the user interface designing. When making user interface design decisions, you should take into asses the physical and mental capabilities of the people who use software. Following are the user interface design principles.

User Familiarity

The interface should use terms and concepts drawn from the experience of the people who use the system

Consistency

The interfaces should be consistent in that wherever possible, comparable operations should be activated in the same way

Minimal surprise

Users should never be surprised by the behaviour of the system

Recoverability

The interfaces should include mechanisms to allow users to recover from errors

User guidance

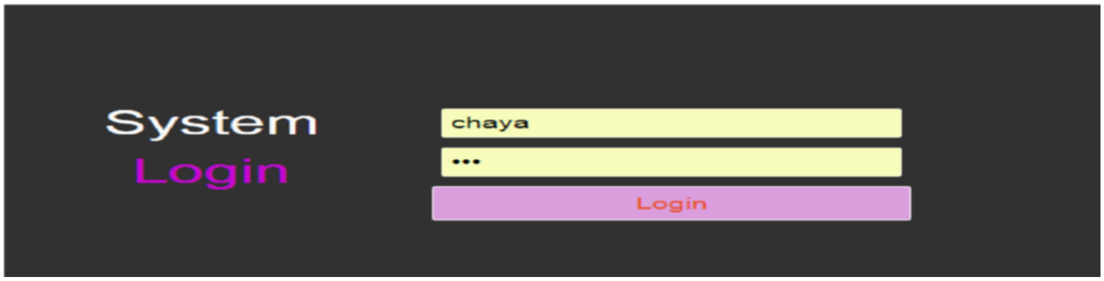
The interface should provide meaningful feedback when errors occur and provide help facility.

User Diversity.

The interface should provide appropriate interaction facilities for different type of user.

3.7.1 Interface of login page.

User should enter a valid username and password in order to access the system. The login interface of the system is depicted in figure 3.5.1



System Login

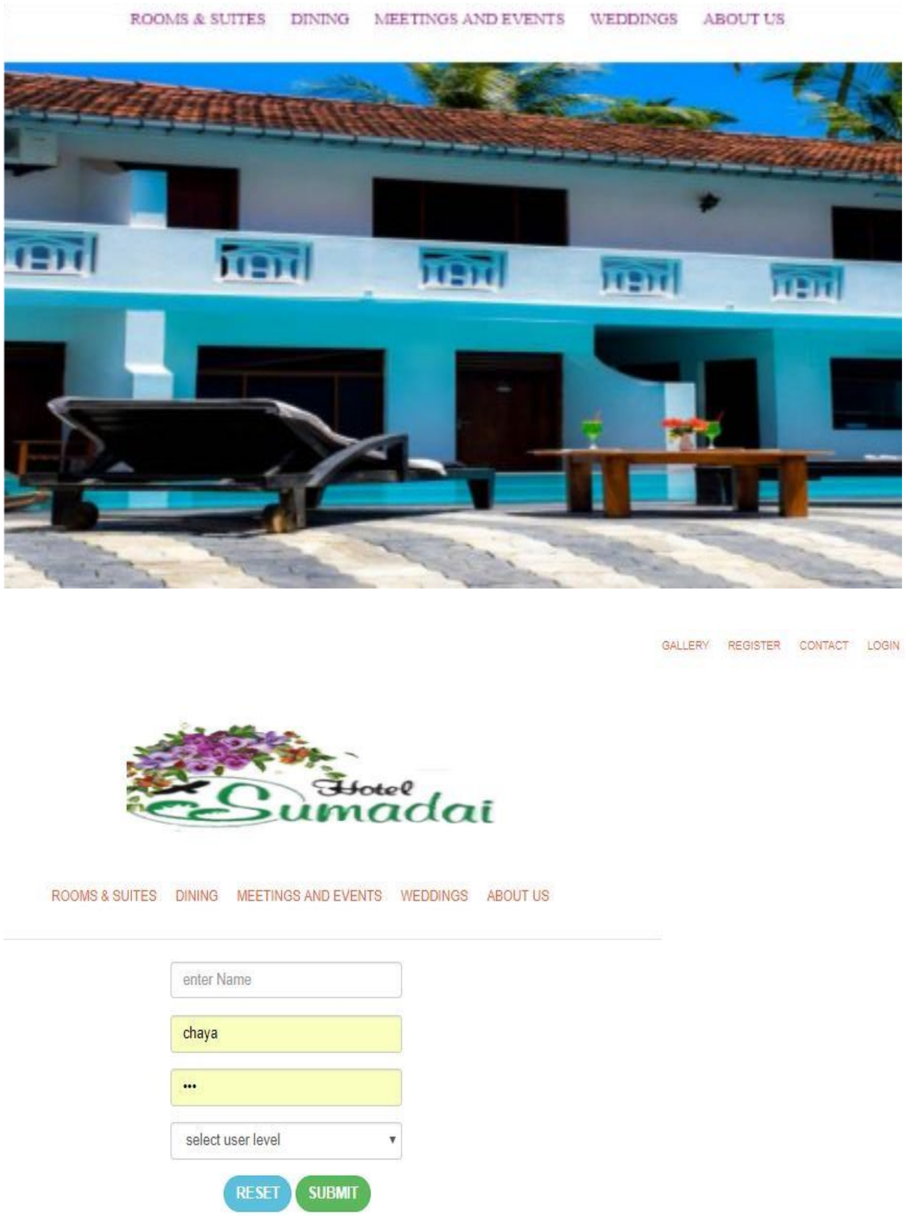
chaya

•••

Figure 3.7.1 Interface of login page

3.6.2 Interface of reservation.

User should enter arrival date, departure date and number of rooms to reserve Room



enter Name

chaya

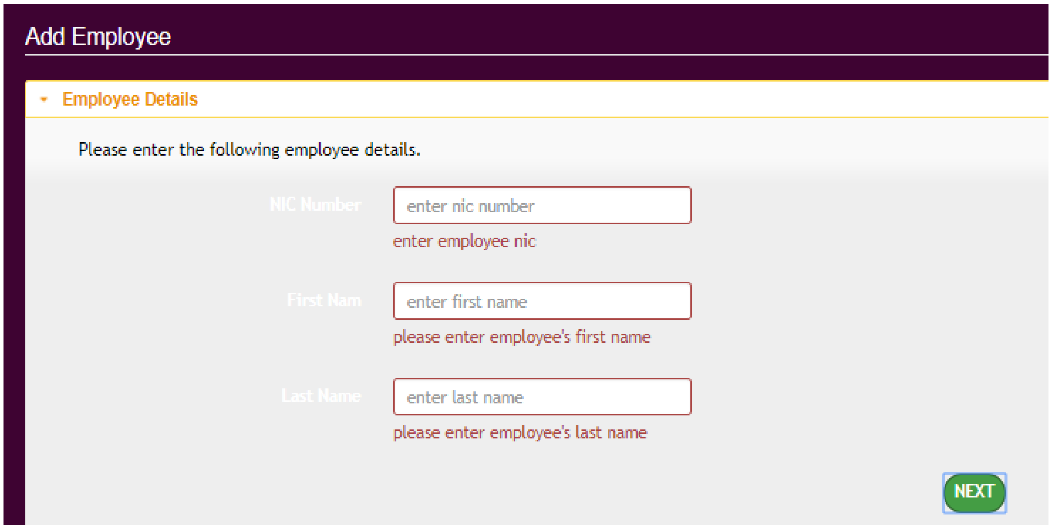
•••

select user level ▾

Figure 3.7.2 Interface of reservation

3.7.3 Interface of add employee details.

Figure 3.7.3 represent the employee registration which can store the details of employee. This form facilitated to store image of the employees.



Add Employee

Employee Details

Please enter the following employee details.

NIC Number: enter nic number

enter employee nic

First Nam: enter first name

please enter employee's first name

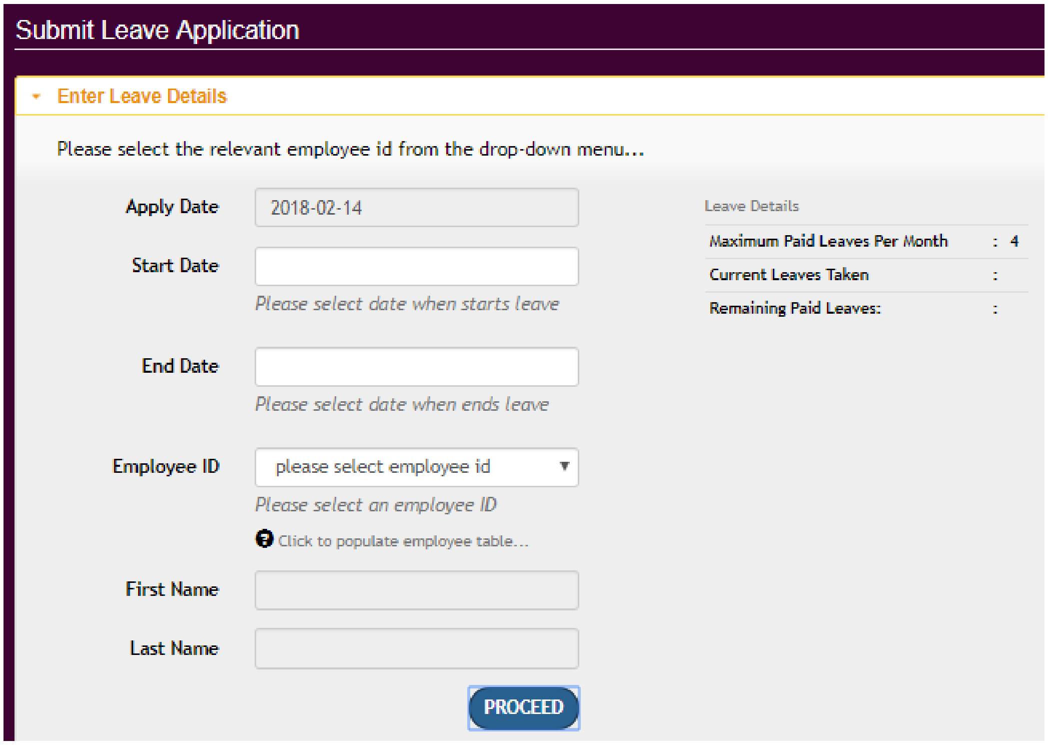
Last Name: enter last name

please enter employee's last name

Figure 3.7.3 Interface of add employee details.

3.7.4 Interface of employee leave form.

User can add leave data in to the system and request for leave.



Submit Leave Application

Enter Leave Details

Please select the relevant employee id from the drop-down menu...

Apply Date: 2018-02-14

Start Date:

Please select date when starts leave

End Date:

Please select date when ends leave

Employee ID: please select employee id

Please select an employee ID

Click to populate employee table...

First Name:

Last Name:

Leave Details

Maximum Paid Leaves Per Month	:	4
Current Leaves Taken	:	
Remaining Paid Leaves:	:	

***Figure 3.7.4 Interface of add employee details*.**

CHAPTER 4
IMPLEMENTATION

4. IMPEMENTATION

4.1 INTRODUCTION

In this chapter it is explained how the clients' requirements that have been analysed and logically designed content is actually converted to a physical system. The development tools, technologies and hardware requirements are also explained in this chapter. Important code parts are also explained in this chapter.

4.2 DEVLOPMENT ENVIRONMENT

Hardware configuration	**Software Configuration**
Intel Pentium Core 3 processor	Xampp sever
8 GB RAM	Php 5+
500 GB Hard Disc with 250 GB free space	My sql 5+
	NetBeans IDE 8.1
	Adobe Photoshop
	Adobe Dream weaver

Table 4.1-Devolopment Software and Hardware Requirement

Client users

Hardware Configuration	**Software configuration**
Intel Pentium Dual Core processor 1 GB RAM	Microsoft Windows and above Any Web Browser
80 GB Hard Disc with 10 GB free space	

Table 4.2-Client software and Hardware Requirement

4.3 REASON TO CHOOSE SOFTWARE

Xampp Server

Is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, Maria DB database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache (A), Maria DB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. [8]

MySQL

MySQL is an open-source relational database management system (RDBMS) The MySQL development project has made its source code available under the terms of the GNU General Public License. MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). [9]

PHP

As the development language of the system PHP was selected. As it is an open source widely used scripting language with platform independence. And as the web server and RDBMS software are perfectly compatible with PHP. [10]

4.4 DEVELOPMENT TOOLS USED

NetBeans IDE

Is a software development platform written in Java. The NetBeans Platform allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans Platform, including the NetBeans integrated development environment (IDE), can be extended by third party developers. [11]

Adobe Photoshop

Adobe Photoshop was used to design the interfaces of the system. It was useful for creating icons and other imagery items. [12]

Adobe Dreamweaver

Adobe Dreamweaver is a WYSWIG code editor for HTML, and supports scripting languages like PHP, JSP, and ASP etc. This tool was used to code and develop the system [13].

4.5 Network Architecture

The Implemented System is a web based system consisting with a web server and database server. Users access the system via a web browser and should have an active internet connection. The servers are protected by a firewall.

4.6 Application Architecture

The system was developed using Model-View-Controller (MVC) architecture. MVC separates information and users' interaction with it. This allows code reusability, easy maintenance and high flexibility.

View

View folder contains all the user interface files. Changes to the user interfaces and all the styles have been given here. It also contains User images, JavaScript files and plugins used for user interface design.

Controller

The Controller acts like an agent between Model and View. Database values are passed to the controller and controller presents it to the view in defined format.

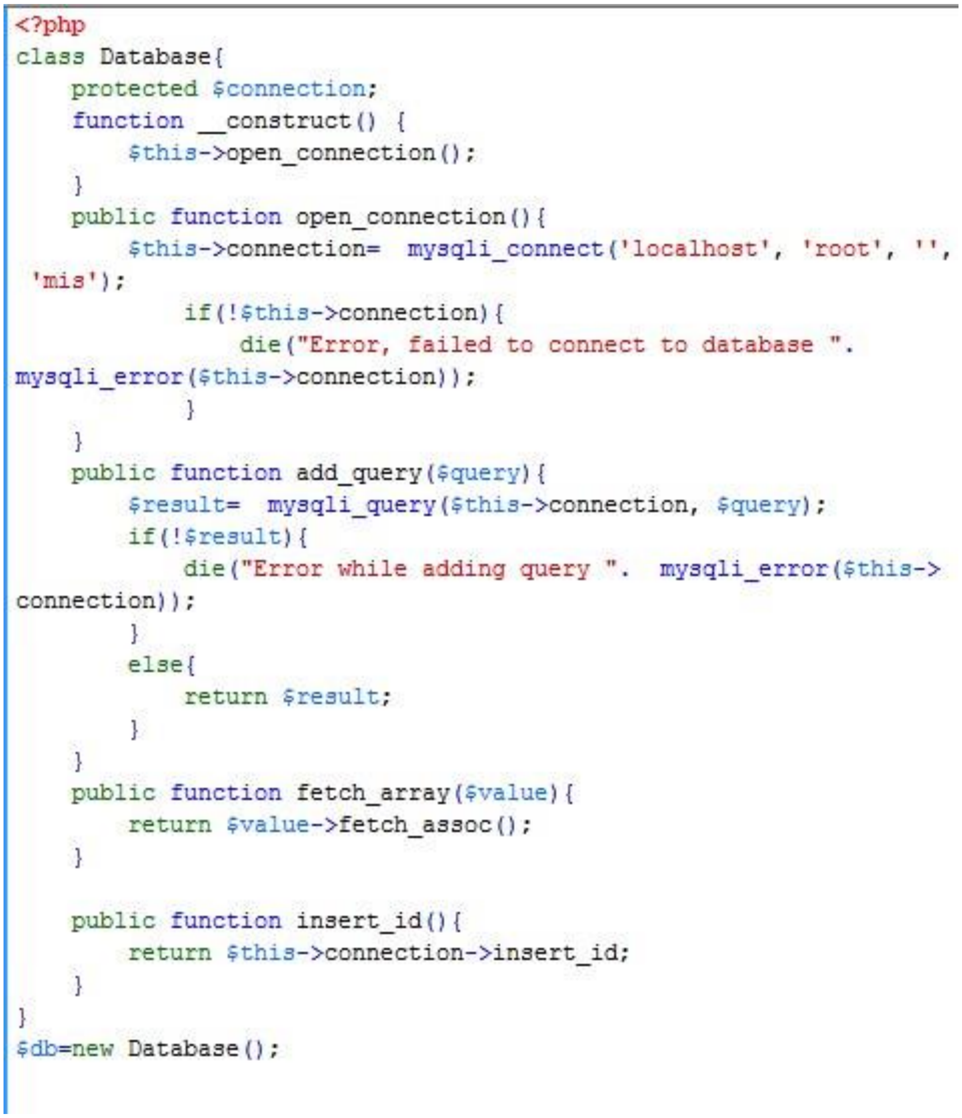
Model

All the database related operations are handled by the Modal. Queries will be created and executed and results will be returned back to the Controller or View.

4.7 MAJOR CODE SEGMENTS

4.7.1 Database Connection

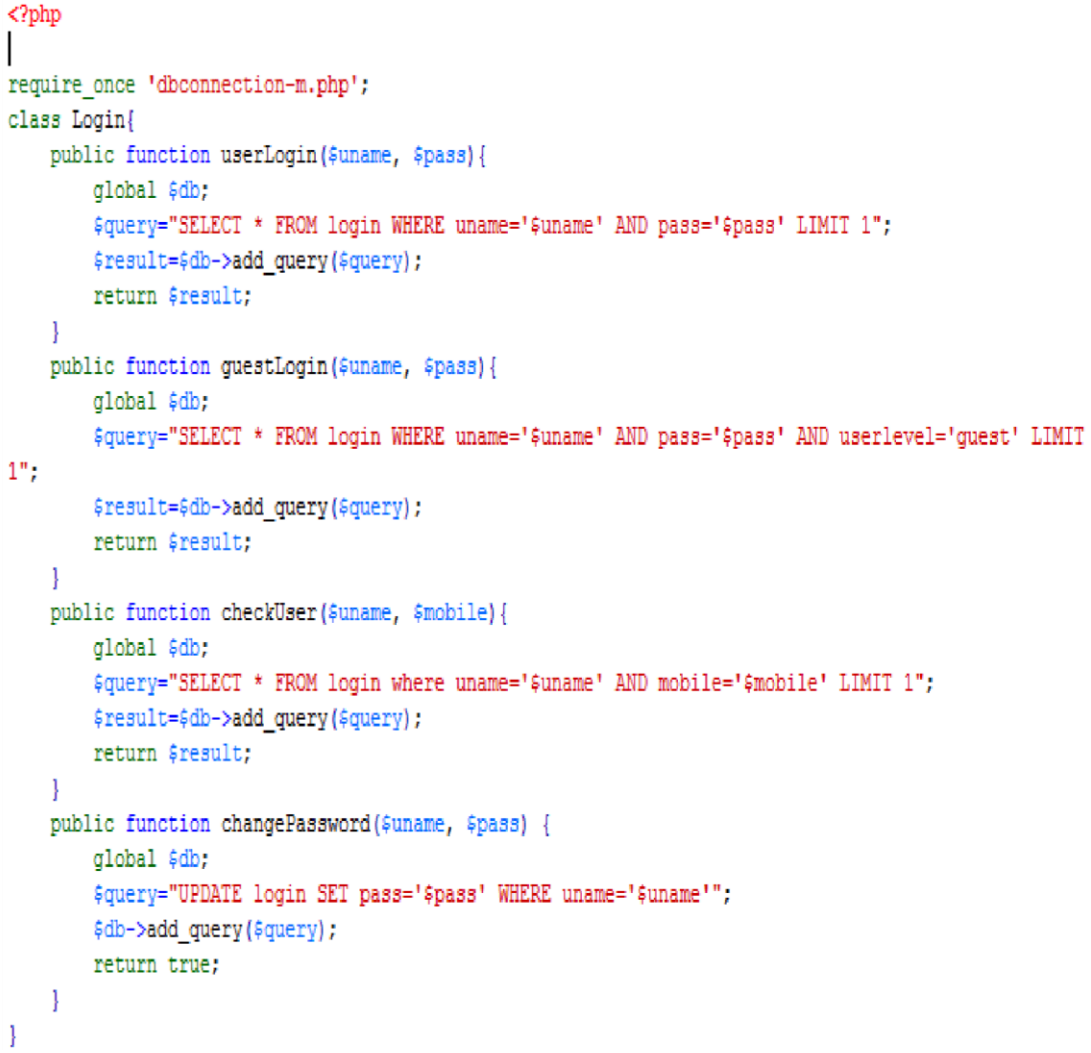
Connection of the database is done on the initiation of the database class. Database Connection is a reusable function. The database connection function is called from the construct method of the class and whenever the class is initiated the database connection is automatically connected.



```
<?php
class Database{
    protected $connection;
    function __construct() {
        $this->open_connection();
    }
    public function open_connection(){
        $this->connection=  mysqli_connect('localhost', 'root', '',
 'mis');
            if(!$this->connection){
                die("Error, failed to connect to database ".
mysqli_error($this->connection));
            }
    }
    public function add_query($query){
        $result=  mysqli_query($this->connection, $query);
        if(!$result){
            die("Error while adding query ".  mysqli_error($this->
connection));
        }
        else{
            return $result;
        }
    }
    public function fetch_array($value){
        return $value->fetch_assoc();
    }

    public function insert_id(){
        return $this->connection->insert_id;
    }
}
$db=new Database();
```

4.7.2 Login to the system

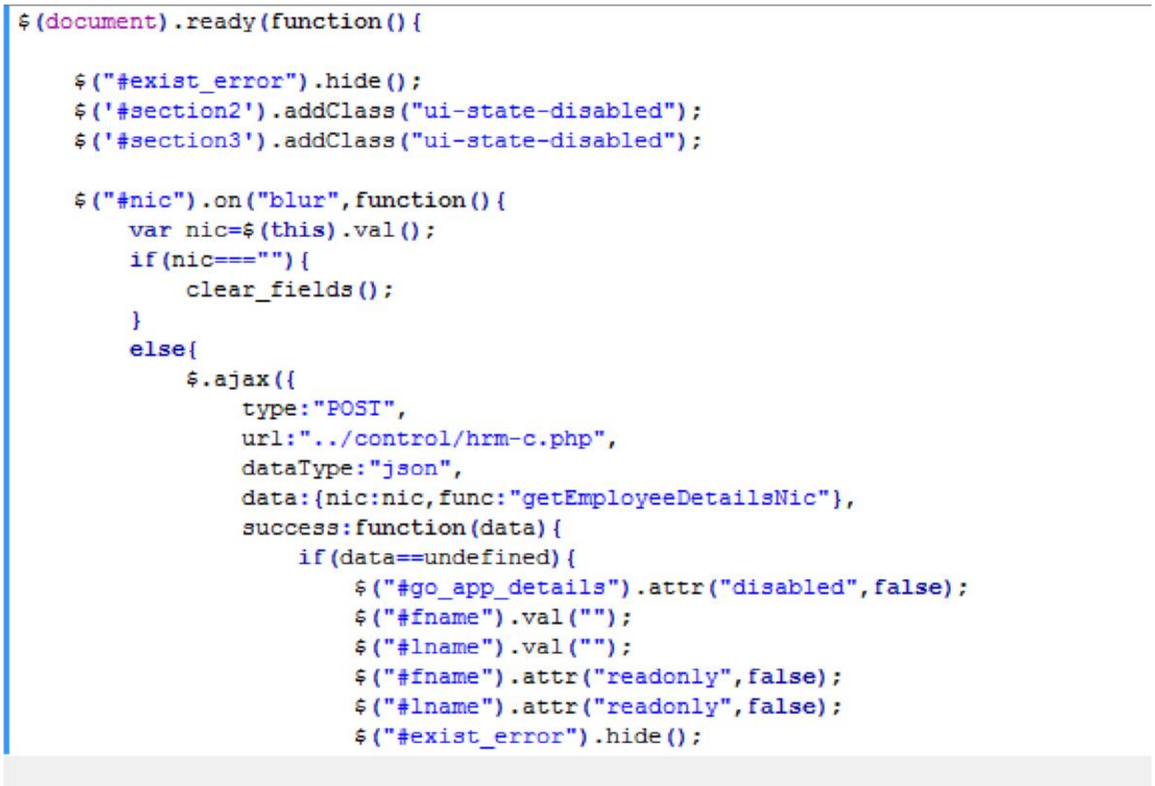
Registered users can access to the system with their username and password. In the View user will be presented with an interface to enter username and password. Upon validation the user entered username and password will be passed to the Controller page named as login. In the Controller login page username and password will be retrieved and will be passed to Modal to validate the user. Depending on the returned results from Modal user will be redirected to login page or index page



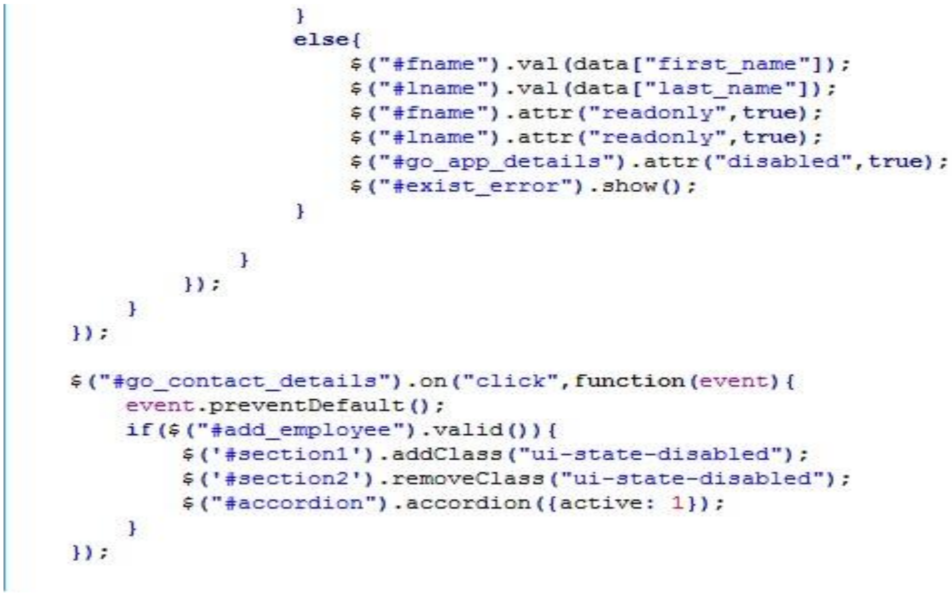
```
<?php

require_once 'dbconnection-m.php';
class Login{
    public function userLogin($uname, $pass){
        global $db;
        $query="SELECT * FROM login WHERE uname='$uname' AND pass='$pass' LIMIT 1";
        $result=$db->add_query($query);
        return $result;
    }
    public function guestLogin($uname, $pass){
        global $db;
        $query="SELECT * FROM login WHERE uname='$uname' AND pass='$pass' AND userlevel='guest' LIMIT
1";
        $result=$db->add_query($query);
        return $result;
    }
    public function checkUser($uname, $mobile){
        global $db;
        $query="SELECT * FROM login where uname='$uname' AND mobile='$mobile' LIMIT 1";
        $result=$db->add_query($query);
        return $result;
    }
    public function changePassword($uname, $pass) {
        global $db;
        $query="UPDATE login SET pass='$pass' WHERE uname='$uname'";
        $db->add_query($query);
        return true;
    }
}
```

4.7.3 Data Validation

Data validation is a most important and critical of the implementation process. Because users can enter unwanted values to the system, so that accuracy of the data can be loosed.



```
$(document).ready(function(){

    $("#exist_error").hide();
    $('#section2').addClass("ui-state-disabled");
    $('#section3').addClass("ui-state-disabled");

    $("#nic").on("blur",function(){
        var nic=$(this).val();
        if(nic===""){
            clear_fields();
        }
        else{
            $.ajax({
                type:"POST",
                url:"../control/hrm-c.php",
                dataType:"json",
                data:{nic:nic,func:"getEmployeeDetailsNic"},
                success:function(data){
                    if(data==undefined){
                        $("#go_app_details").attr("disabled",false);
                        $("#fname").val("");
                        $("#lname").val("");
                        $("#fname").attr("readonly",false);
                        $("#lname").attr("readonly",false);
                        $("#exist_error").hide();
```



```
                    }
                    else{
                        $("#fname").val(data["first_name"]);
                        $("#lname").val(data["last_name"]);
                        $("#fname").attr("readonly",true);
                        $("#lname").attr("readonly",true);
                        $("#go_app_details").attr("disabled",true);
                        $("#exist_error").show();
                    }

                }
            });
        }
    });

    $("#go_contact_details").on("click",function(event){
        event.preventDefault();
        if($("#add_employee").valid()){
            $('#section1').addClass("ui-state-disabled");
            $('#section2').removeClass("ui-state-disabled");
            $("#accordion").accordion({active: 1});
        }
    });
```

CHAPTER 5
EVALUATION

5. EVALUATION

5.1 INTRODUCTION

The objective of this chapter is to give the reader an understanding about the testing phase of the proposed system. This chapter will focus on the testing plan, testing schedule, test cases, screen shots of the error messages and user acceptance testing.

5. 2 TESTING METHODS

There is various kind of testing methods available to understand the strengths and weaknesses of the system. White box testing and Black box testing are two main testing methods used in software testing.

5.2.1 WHITE BOX TESTING.

White box testing is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit. [14]

5.2.2 BLACK BOX TESTING

Black Box Testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing. [15]

5.3 TEST PLAN

5.3.1 UNIT TESTING

A unit is a relatively small piece of code consisting with little functionality. The intension of unit testing is to check whether we are getting the intended output. Unit testing can be a black box test or a white box test. [16]

5.3.3 INTEGRATION TESTING

Unit tested individual code fragments have to be integrated to ensure they work well when integrated. Sometimes individual code fragment may work fine but when integrated with other modules it may give errors. The purpose of integration testing is to identify those issues with integration [17]

5.3.4. REGRESSION TESTING

Software usually evolves over the time and changes made to the system should not introduce new errors. Regression testing aims at identifying problems arisen with the changes made to the system. [18]

5.3.5. SYSTEM TESTING

After successful completion of unit and integration testing, the whole system has to be tested in a real environment similar to the future working environment. System testing aims at identifying problems arisen in real working environment. [19]

5.3.6. USER ACCEPTING TESTING

A system should be tested by the end users and should verify whether the intended functionalities specified in specification were been implemented. Acceptance from the user indicates the system has met its objectives. HMS was tested in real working environment by the end users with different user levels and privileges. Users were impressed with the system and most of them gave positive feedback. Finally, it was decided that the HMS was satisfying and the system was accepted by the end users.

5.4 TEST CASES

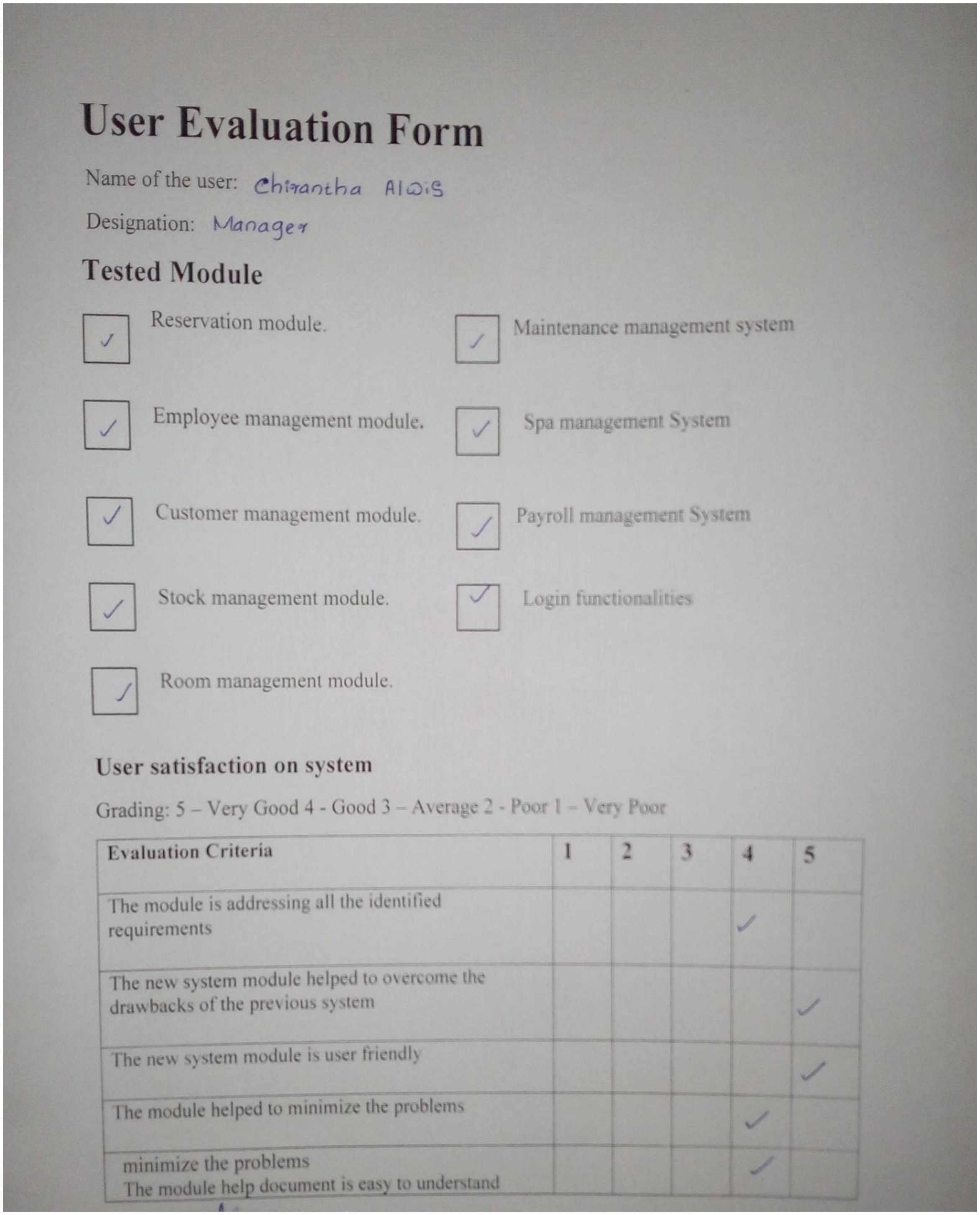
Module name	**Function name**	**Test priority**
Reservation module	Insert reservation details.	High
	Delete reservation details	High
	Update reservation detail	High
	Confirm deletion detail of reservation	High
	View reservation detail.	High
	Search reservation detail	Low
	Insert credit card details.	High
Employee information management module	Insert employee details	High
	Delete employee details.	High

	Update Employee details.	High
	View employee details.	High
	Search employee details by name	Low
	Search employee details by employee id.	Low
	Apply for leave	High
	Approve for leave	High
	Add employee attendance detail.	High
Customer information management module	Add customer details	High
	Update customer details	Medium
	Delete customer details.	High
	View customer details.	High
	Add billing data of customer	High
	Print bill for customer	High
	Search customer by name	Low
	Search customer by id	Low
Stock management module	Add new stock	High
	Update new stock	High
	Delete new stock	High
	Check stock quantity	Medium
	View stock details	High
	Status of quantity display with colour	Medium

Manage room inventory module	Add room status	High
	Update room status	High
	Delete room status	High
	View room status	High
	Check room status	Medium
	Add room price	High
	Update room price	High
	Book rooms	Medium
	View room information	High
	View reservation information	High
Massage alert notification	Sent message and email to hotel About reservation details.	High
Maintain management module	Add maintain detail	High
	Update maintain detail	High
	Delete maintain detail	High
	View maintain detail	High
Payroll management module	Calculate salary	Medium
	View pay slip	High
	Print pay slip	High

Table5.5 1.-Test Case Result

USER EVALUATION FORM



User Evaluation Form

Name of the user: Chirantha Alwis

Designation: Manager

Tested Module

☑ Reservation module.

☑ Employee management module.

☑ Customer management module.

☑ Stock management module.

☑ Room management module.

☑ Maintenance management system

☑ Spa management System

☑ Payroll management System

☑ Login functionalities

User satisfaction on system

Grading: 5 – Very Good 4 - Good 3 – Average 2 - Poor 1 – Very Poor

Evaluation Criteria	1	2	3	4	5
The module is addressing all the identified requirements				✓	
The new system module helped to overcome the drawbacks of the previous system					✓
The new system module is user friendly					✓
The module helped to minimize the problems				✓	
minimize the problems The module help document is easy to understand				✓	

CHAPTER 6
CONCLUSION

6. CONCLUSION

6.1 INTRODUCTION

Hotel Sumadai is well known hotel in Beruwala. They need to enhance their business with new technology. Earlier they faced many problems with manual system. But they are carrying out their reservation and day to day activities effectively. Earlier they faced many difficulties of handling room reservation, Maintenance management and handling customer and employee details. They have to spend to lot of time handle this process.

Currently because of this hotel management system, they can carry out their activities effectively and efficiently. HMS project has a simple and user-friendly interfaces they can do their work very easy.

I can happy to say this and I am proud of myself because I have completed the system successfully and have acknowledged .net, Dream waver and Xampp Server in to considerable extends. In addition to this was a great opportunity for me to get practical experience of how to build large software product and how to manage software project successfully. Time management of the project is more important. I have successfully managed the time during the period of year. In addition, course units I followed during last three years were behind of my success among those course system Software Engineering, Database management, Project Management and Rapid Application Development were important.

6.2 CRITICAL ASSESSMENT OF THE PROJECT

The system was proposed to help to manage the operations of hotel management system efficiently & effectively. As the very first step of the project a feasibility study was carried out in order to ensure the proposed system is worth enough to develop. Then the system analysis was done and collected the requirements. Different fact finding techniques were used to identify the requirements. Based on the identified requirements system design was done according to the RUP methodology and Object Oriented approach with the support of few designing tools. Identified requirements were reviewed regularly to ensure the accuracy of the gathered requirements. php was used to develop the system with the . NetBeans IDE 8.1 while Microsoft SQL Server 2005 was used to develop the database. System testing was started in the development face carried out throughout the development phase and evaluation phase according to the test cases developed in design phase. Test cases and the expected result is documented and compared with the actual results to ensure that the system performs its functions as expected. Useful features which gives a lot of control over the stock. Some of those features Such as create purchase orders, basic operations of return management, reports to check the inventory level are included in the hotel Management System while the features like e-mail facility, customizable databases are not available.

6.3. LESSONS LERNT

Final year project was a great learning curve and experience in real life software development. Theories and principles were put in to use and industry best practices were adhered. It gave me a wonderful opportunity to be involved in a complete software development life cycle. It provided a great deal of self-learning and research about the latest technologies and practices. Technologies such as AJAX and jQuery were totally new and using them with PHP server side scripting provided a great deal of self-learning. Front end UI design frame works such as jQuery UI and Bootstrap were learnt from the beginning through the implementation.

6.4. FUTURE ENHANSEMENT

Following are some of the future enhancements the client willing to have. Introducing online reservation through mobile.

6.4 REFFERENCES

[1]-SIMILAR SYSTEM
https://www.youtube.com/watch?v=rUE4US0UgQo[14.04.2017]

[2]- SIMILAR SYSTEM
https://www.youtube.com/watch?v=A1qy77yKe34 [14.04.2017]

[3]- SIMILAR SYSTEM
http://www.findapp.com/apphp-hotel-site-web-reservation-system-d52916.html
[14.04.2017]

[4]- SIMILAR SYSTEM
https://itsourcecode.com/2016/01/complete-hotel-management-system[14.04.2017]

[5]-water fall model
https://en.wikipedia.org/wiki/Waterfall_model [14.04.2017]

[6]-Prototyping

https://en.wikipedia.org/wiki/Software_prototyping[16.04.2017]

[7]-Rational Unified model
https://en.wikipedia.org/wiki/Rational_Unified_Process [14.04.2017]

[8]-Xampp sever
https://en.wikipedia.org/wiki/XAMPP [10.05.2017]

[9]-MYSQL
https://en.wikipedia.org/wiki/MySQL [10.05.2017]

[10]-PHP
https://en.wikipedia.org/wiki/PHP [10.05.2017]

[11]-NETBEANS
https://en.wikipedia.org/wiki/NetBeans[14.05.2017]

[12]-Adobe photo shop
https://en.wikipedia.org/wiki/Adobe_Photoshop [14.05.2017]

[13]-Adobe Dream Weaver

https://en.wikipedia.org/wiki/Adobe_Dreamweaver [15.05.2017]

[14]-White Box Testing
https://en.wikipedia.org/wiki/White-box_testing [12.06.2017]

[15]-Black Box Testing
https://en.wikipedia.org/wiki/Black-box_testing [12.06.2017]

[16]-Unit Testing
https://en.wikipedia.org/wiki/Unit_testing [12.06.2017]

[17]-Integration Testing
https://en.wikipedia.org/wiki/Integration_testing [14.07.2017]

[18]-Regression Testing
https://en.wikipedia.org/wiki/Regression_testing [14.07.2017]

[19]-System Testing
https://en.wikipedia.org/wiki/System_testing [14.07.2017]

APPENDIX A-

SYSTEM DOCUMENTATION

INSTOLLING THE OPERTING MANAGEMENT SYSTEM

- Install the necessary software
- Download and install XAMPP server in the server machine. You can choose WAMP or LAMP instead of XAMPP depending on the server operating system Download and install a suitable web browser.
- Download and install Ozeki SMS server.

2.Set up OMS

Open "*c:\xampp\htdocs*" folder in XAMPP and copy paste the HMS folder in the given CD-ROM. (if it is WAMP open "*c:\wamp\www*" and for LAMP open "*/opt/69amp/htdocs/*")

3.Install the database.

- Open XAMPP control panel in the desktop by double clicking it.
- Press start button in front of apache module and MySQL module
- Press Admin in MySQL module or type "*http://localhost/phpmyadmin*" in the browser window and open phpMyAdmin.
- Create a new database named as "hotelmanagement" with the collation .
- Select the "hotelmanagement" database from the database list and click on the import button under the database.
- Browse and select the "hotelmanagement.sql" file from the CD-ROM and click the go button.

CONFIGURATIONS

- User can take back up to the default folder and also to a user specified external device.
- To change the default folder location,
- Go to “*C:\xampp\htdocs\HMS\Controller\backup.php”*
- Change the default path .
- To change the backup location click the choose folder button in backup form and select the path you want to save the file.

LAUNCHING THE HMS

After the successful installation of HMS and the database open the preferred web browser and type “http://localhost/HMS/login’”and the log in page of the system will be loaded.

You can bookmark the mentioned URL above and drag and drop that bookmark in to the desktop for ease of access.

APPENDIX B-

DESIGN DOCUMENTATION

USE CASE DIAGRAM AND DESCRIPTION

- Use case diagram of reservation module

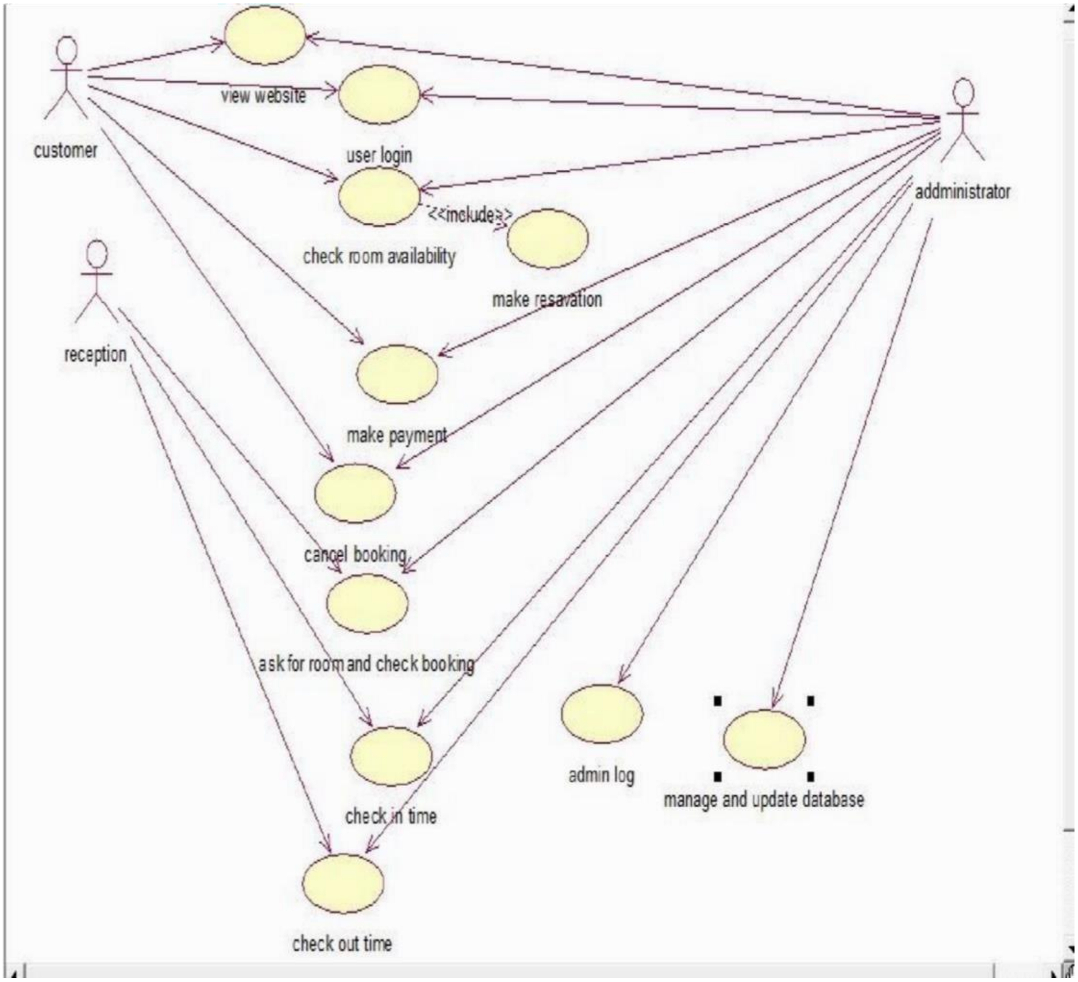


Figure B.1: Use case diagram of the Reservation module

Use case name	**Make reservation**
Actors	Customer, administrator, reception
Description	Make reservation based on the customer and check availability of system.
Precondition	Hotel should have available room

Typical course of event	**Actor action**	**System response**
	1.Select arrival and departure date and number	
	2.Click next button	
		3Display room details.
	4.Select room type	
		5. display error message
		6. Display customer detail page.
	7.Add customer details	
		8. Display customer details.
	9.Add online payment details	
		10. Display successful message.
Alternative courses	5.If that kind of room not available, then display error message or 6.Room is available, Display customer detail page.	
conclusion	This use case emphasizes when reservation process completed data are successfully stored in the database	
Post condition	Reservation data save in the database.	

Table B.1: Use case description for reservation use case.

Use case name	Make payment	
Actors	Customer, admin, reception	
Description	Customer, admin or reception check the availability when make reservation.	
Pre-condition	System users should logged in to the system	
	Actor action	**System response**
	1.Select room type	
		2. Display reservation detail and customer detail form.
	3.Click next button	
		4. Display payment detail form.
	5.Select payment type	
	6.Fill the payment details	
		7. Display message you payment is done.
Alternative courses	System display error message.	
conclusion	This use case emphasizes when customer given the payment detail successfully	
Post condition	Payment detail save in the data base.	

Table B.2: use case description for make payment use case.

USE CASE DIAGRAM FOR CUSTOMER MANAGEMENT MODULE

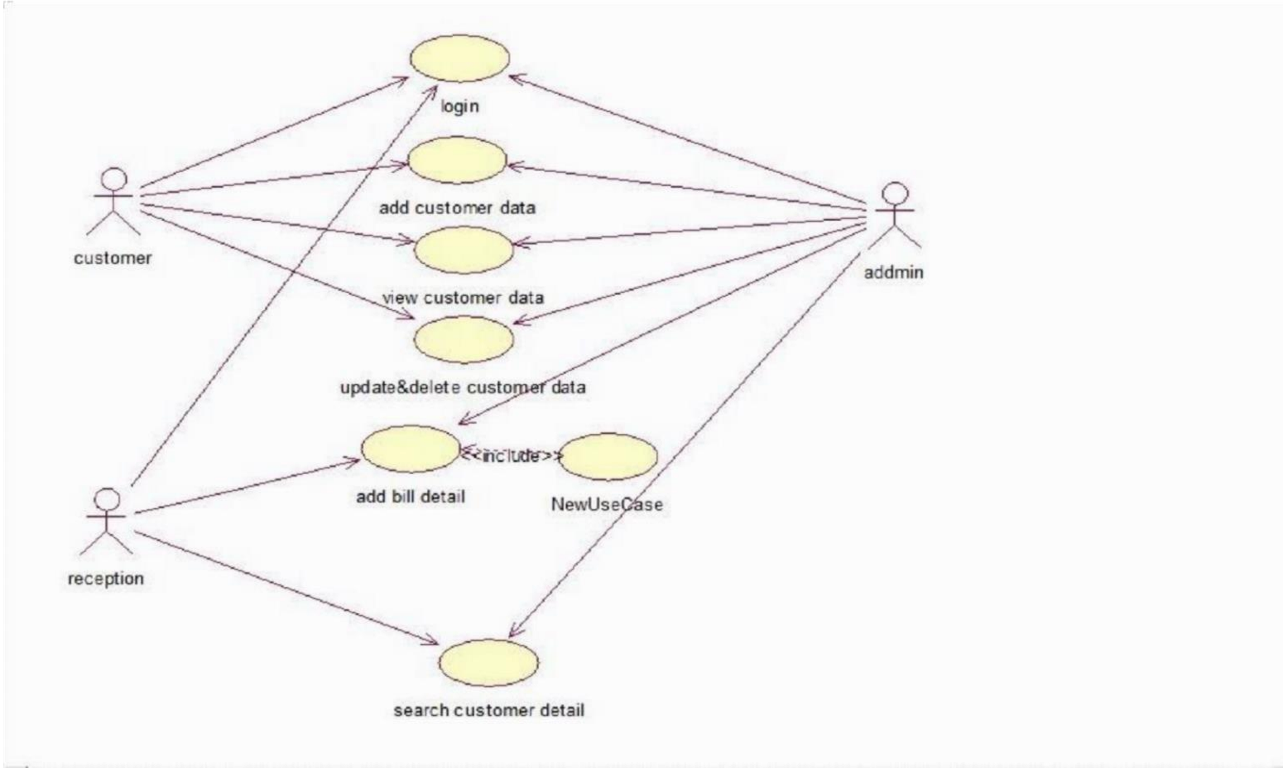


Figure B.2: Use case diagram of the Customer management module

Use case name	Add customer data	
Actors	Customer, reception, admin	
Description	Customer, reception or admin insert data	
Pre-conditions	System users logged in to the system.	
Typical course of events	**Actor action**	**System response**
	Enter the personal details	
	Select male or female	
	Click save button	

		Display next page
Alternative courses	System display error message	
conclusion	This use case emphasizes insertion of the Customer details.	
Post conditions	The data saved in the data base.	

Table B.3: use case description for add customer data use case.

Use case name	Add bill detail	
Actors	Reception, admin	
Description	Receptionist or admin add the data	
Pre-condition	System users should be logged in to the system	
Typical course of events	Actor action	System response
	1.Add customer id, arrival and departure date	
	2.Select type of accommodation	
		3.Display accommodation type
	4.Select payment option	
		5.Display payment option
	6.Click billing button	
		7.Display billing data
Alternative course	7. System display error message.	
conclusion	This use case emphasize insertion of the billing data	
Post conditions	The data saved in the data base.	

Table B.4: use case description for add billing use case.

Use case name	Search customer details.	
Actors	Reception, Admin.	
description	Reception and admin search data of employee	
precondition	System users should be logged in to the system.	
Typical course of event	Actor action	System response
	Entre the customer id	
	Click search button	
		Display customer details.
	Entre the customer name	
	Click search button	
		Display customer details.
Alternative courses	System should display error message	
conclusion	This use case emphasizes insertion of the customer id or name	
Post condition	The data are searched from data base.	

Table B.5: use case description for search customer use case.

USE CASE DIAGRAM FOR EMPLOYEE MANAGEMENT MODULE

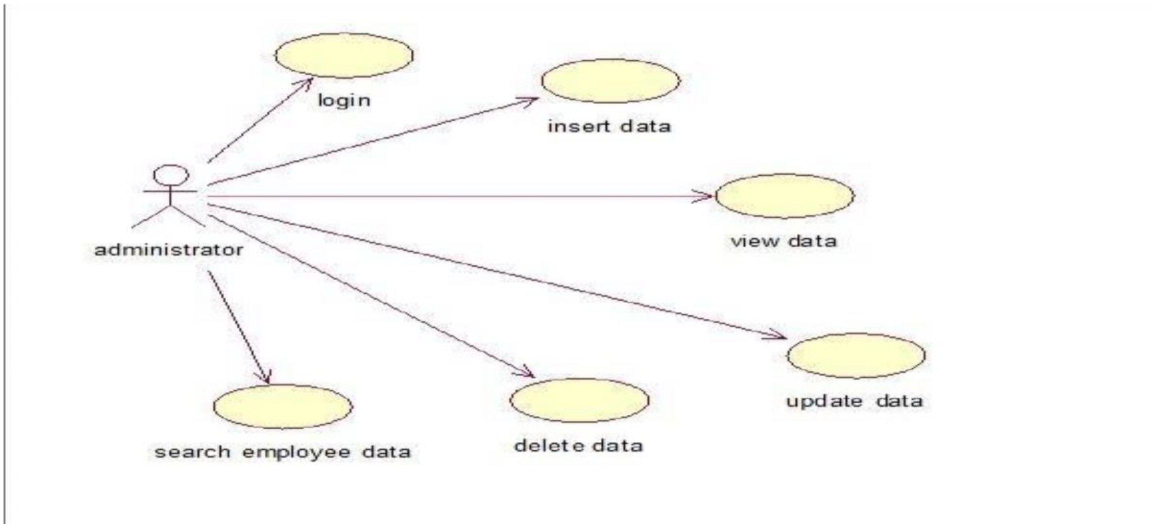


Figure B.3: Use case diagram of the Employee Management module

Use case name	Add employee detail	
Actors	admin	
Description	Admin entre the employee details	
Precondition	System user should logged in to the system	
Typical course of event	**Actor action**	**System response**
	1.Enter full name, employee id, date of birth, nic_no, epf_no , post, salary. and personal address	
	2.Select title	
		3.Display title
	4.Enter nationality	
	5.Select civil status	
		6.Display civil status
	7.Add photograph, telephone number and email address	
	8.Click save button	

		9.Display successfully add message
Alternative course	9. The system display error message.	

conclusion	This use case emphasizes insertion of add employee data.
Post condition	The data are saved in the data base.

Table B.6: use case description for add employee detail use case.

USE CASE DIAGRAM FOR STOCK MANAGEMENT SYSTEM.

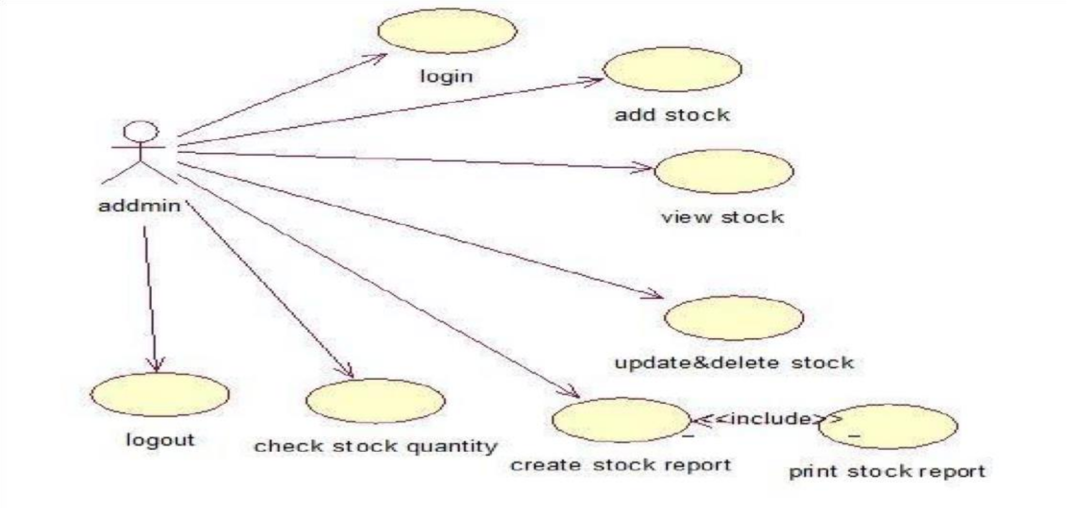


Figure B.4: Use case diagram of the Stock Management module

Use case name	Check stock quantity	
Actors	Admin	
Description	Admin can get all details of stock.	
Pre-condition	users should be logged in to the system	
Typical-course of event	**Actor action**	**System response**
	enter the stock details	
	Click save button	
		Show successfully add message
	Click check stock button.	

		Display stock quantity with colours.
		Auto calculate available stock quantity.
Alternative Courses	System display available Quantity.	
Conclusion	This use case emphasizes the stock available quantity.	
Post-Condition	The system display the data in data table.	

Table B.7: use case description for add stock detail use case.

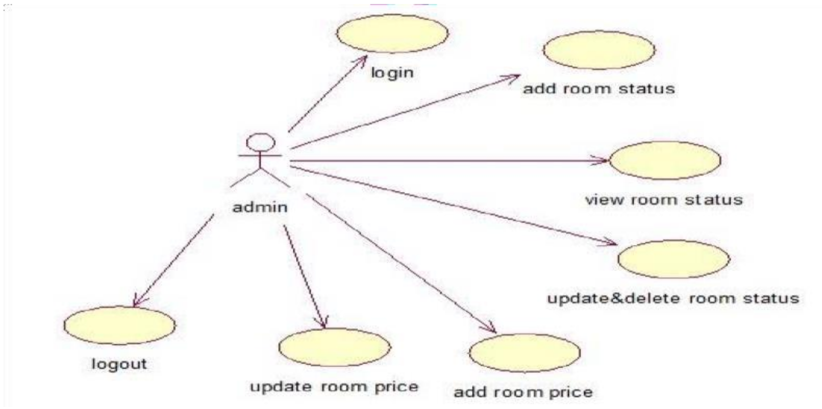


Figure B.5: Use case diagram of the Room Management

Use case name	View room status	
Actors	admin	
Description	Admin view a room status	
Pre-condition	Users should be logged in to the system.	
Typical course of event	**Actors action**	**System response**
	1.Enter the room data	
	2.Click save button	
		3.Display successfully add message
	4.Click view room status button	
		5.Display room status.
		6.Display all room details.
Alternative courses	5. Display room status.	
Post-condition	The data are display in table.	

Table B.8: use case description for view room status use case.

USE CASE DIAGRAM FOR PAYROLL MANAGEMENT SYSTEM.

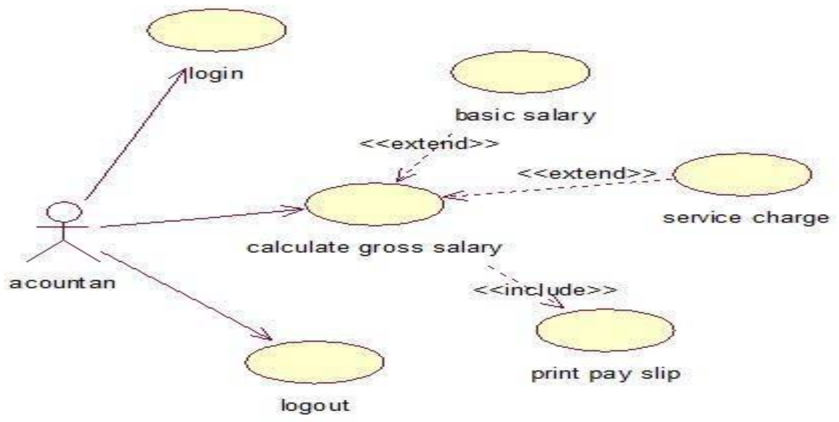


Figure B.6: Use case diagram of the payroll management module

Use case name	Calculate gross salary	
Actors	Accountant	
Description	Users calculate the employee salary	
Pre-condition	Users should be logged in the system	
Typical case of event	**Actor action**	**System response**
	1.Enter basic salary	
	2.Enter service charge	
	3.Click total button	
		4.Display pay slip
Alternative courses	4.Display error message	
conclusion	This use case emphasizes the calculate employee salary.	
Post condition	The play slip must be print.	

***Table B.9: use case description for calculate gross salary use case*.**

Sequence diagram

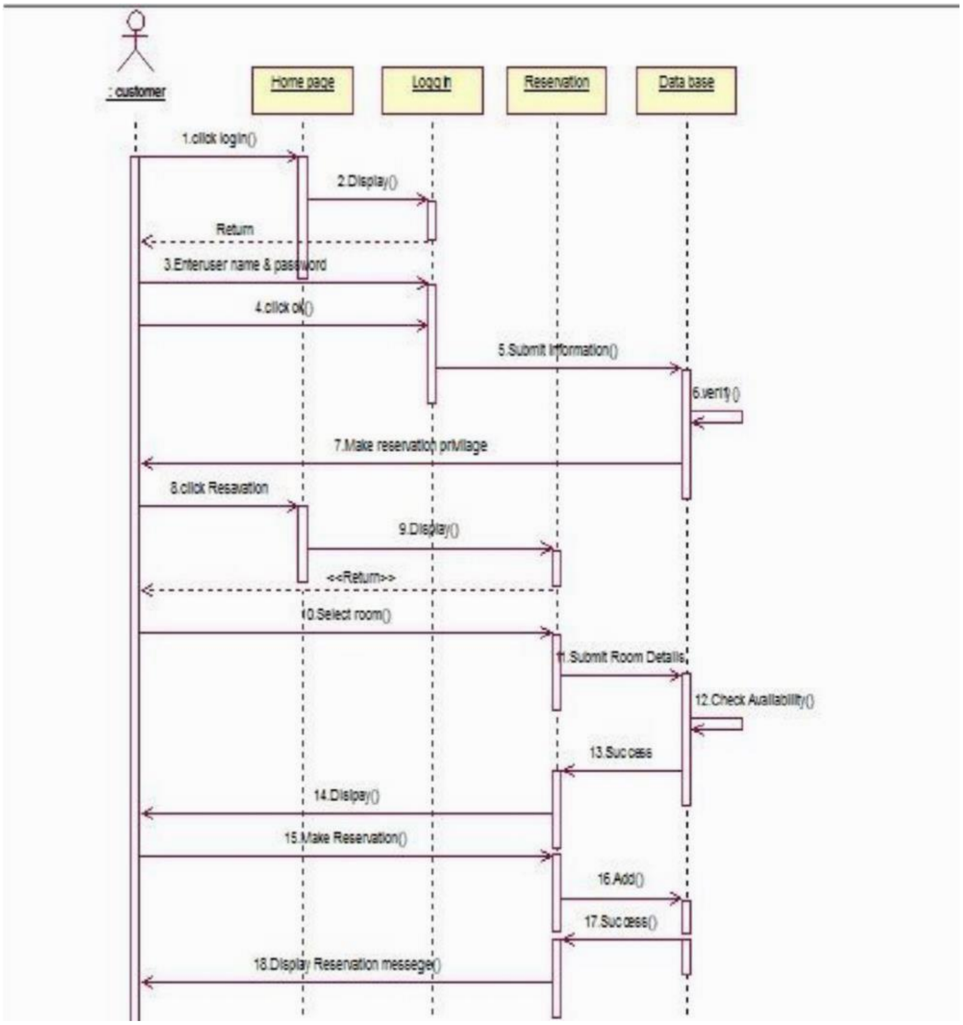


Figure B.7: Sequence diagram of the Reservation module

Activity Diagram

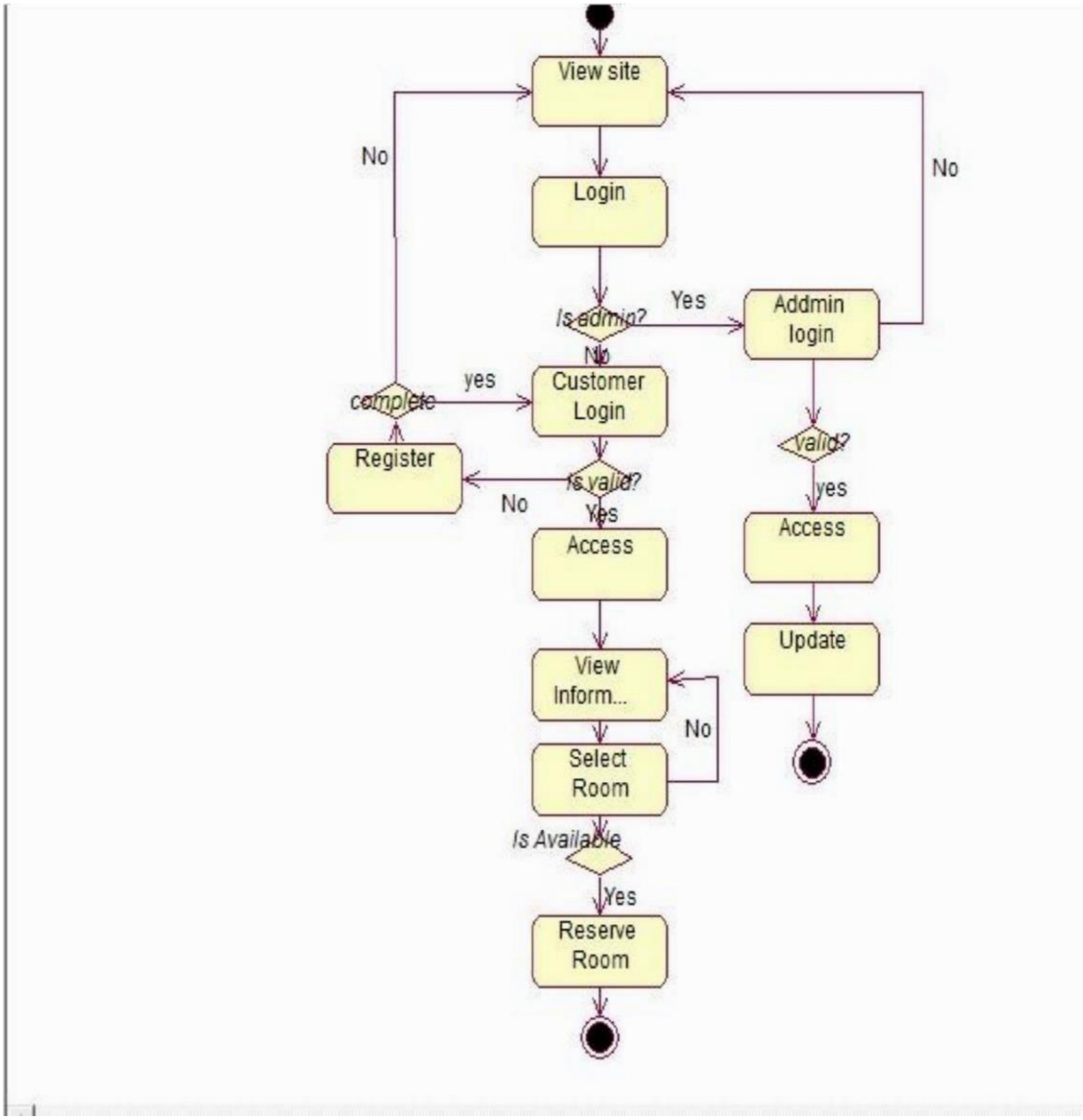


Figure B.8: Activity diagram of the Reservation module.

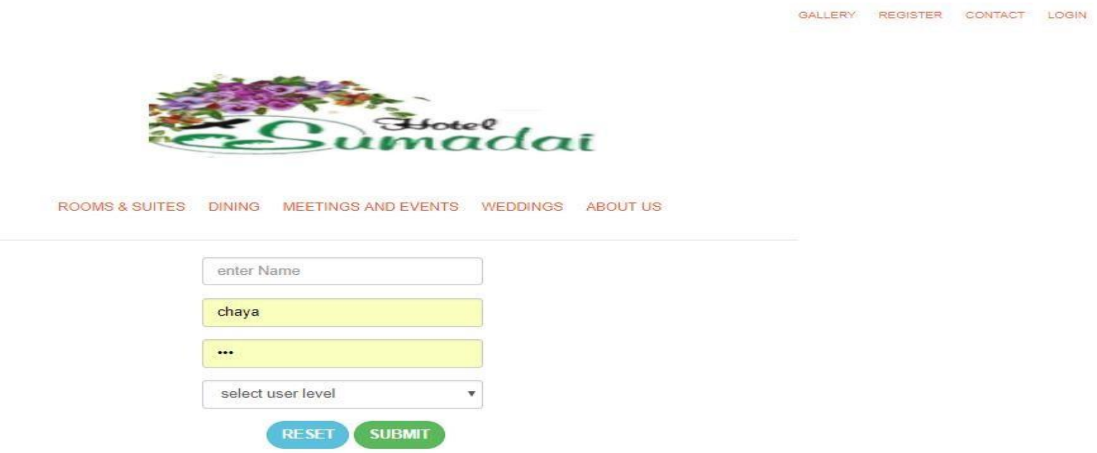
APPENIX C-

USER DOCUMENTATION

C.1 Login form

The user can enter the given user name and the password to login to the system. Type the User name and the Password and Login.

When user try to log into the system check whether this user is valid user or not otherwise system display error message.



chaya

•••

FigureC.1: Login Form

C.2 Main menu items

All the functions of each and every module can be accessible by users through this main form.

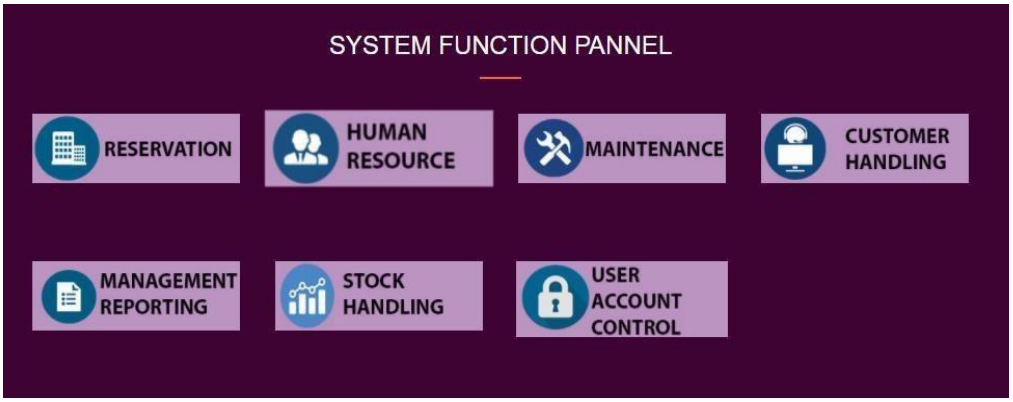


Figure C.2: main Form

C.3 Main menu items of Reservation.

Shows the main menu items of main reservation form

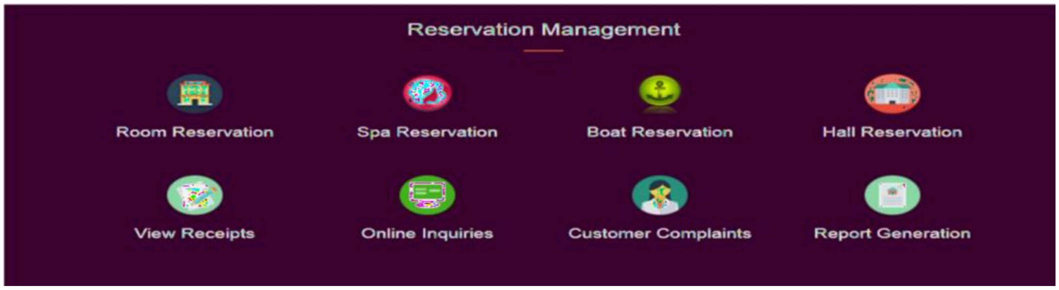
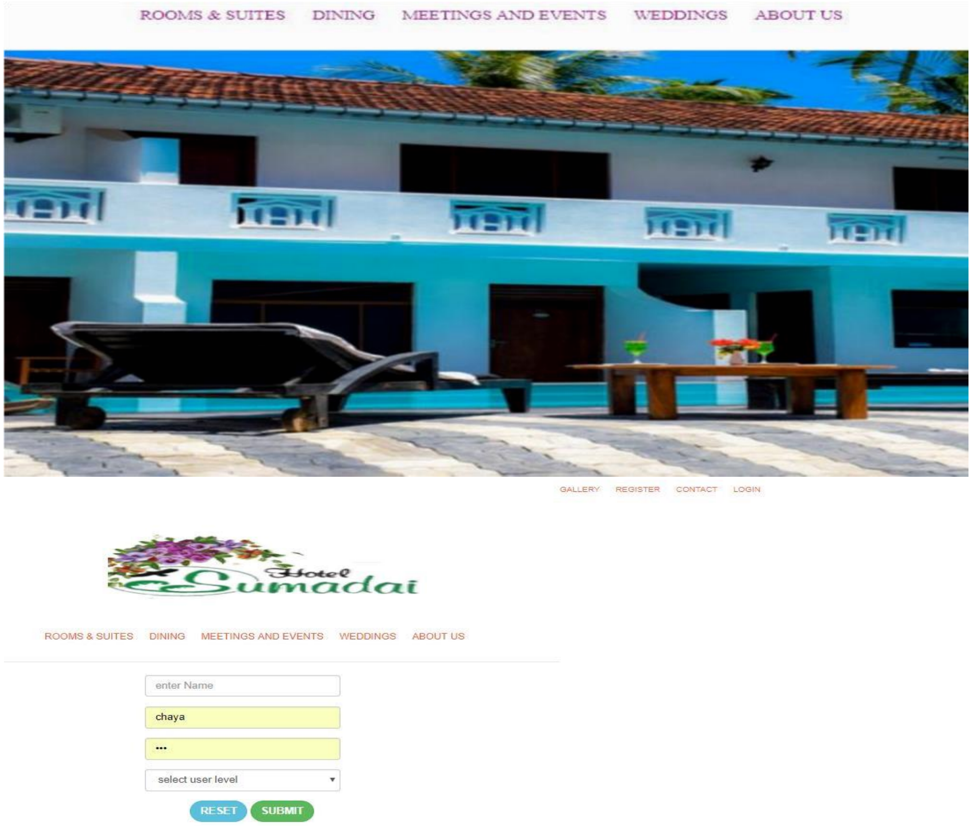


Figure C.3: main Form

C.4 Reservation form

1. Check availability.
2. Select arrival date, departure date and number of rooms.



enter Name

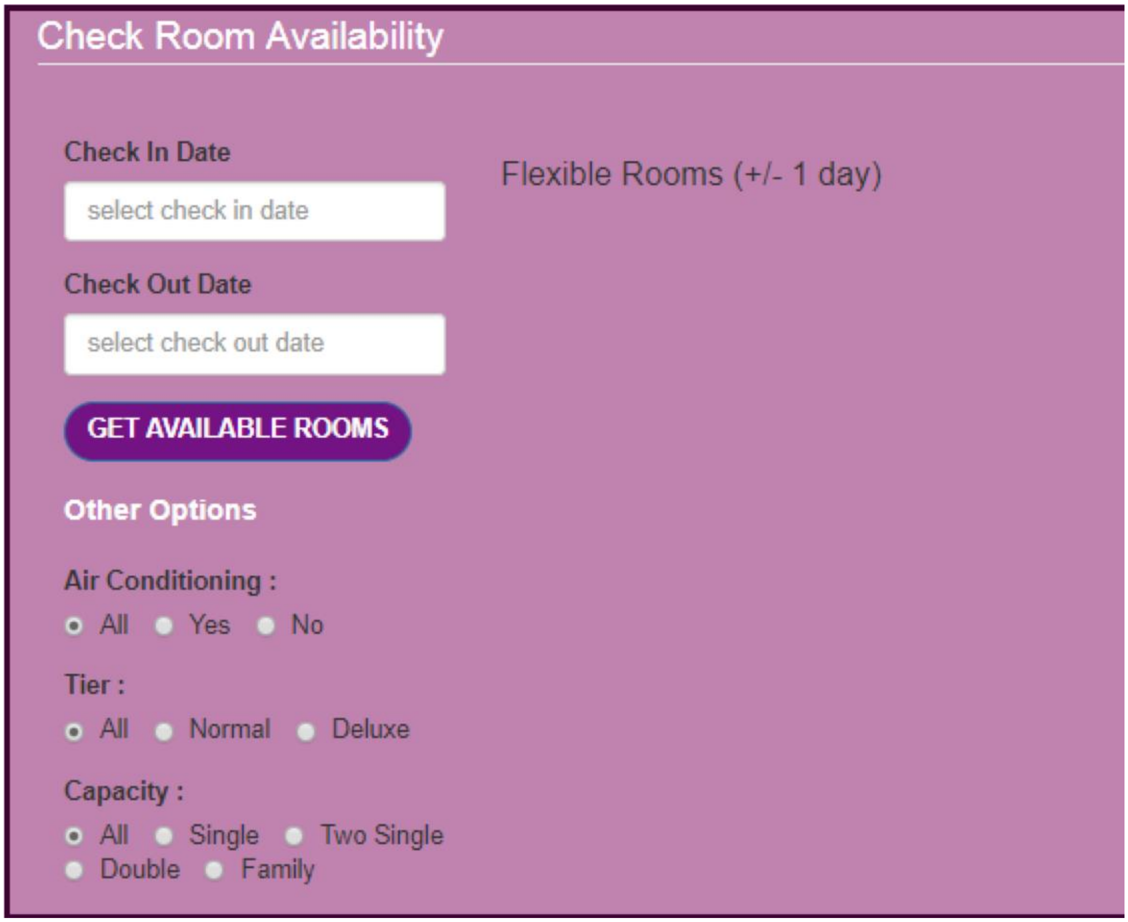
chaya

•••

select user level

Figure C.4: reservation

3. Click main tab



Check Room Availability

Check In Date

select check in date

Check Out Date

select check out date

GET AVAILABLE ROOMS

Flexible Rooms (+/- 1 day)

Other Options

Air Conditioning :

● All ○ Yes ○ No

Tier :

● All ○ Normal ○ Deluxe

Capacity :

● All ○ Single ○ Two Single
○ Double ○ Family

Figure C.5: Booking Form

4. Select room type.
5. Click main menu.

Then display customer details table.

6. Add customer detail and click next button. Then display reservation table.
7. Add reservation details.

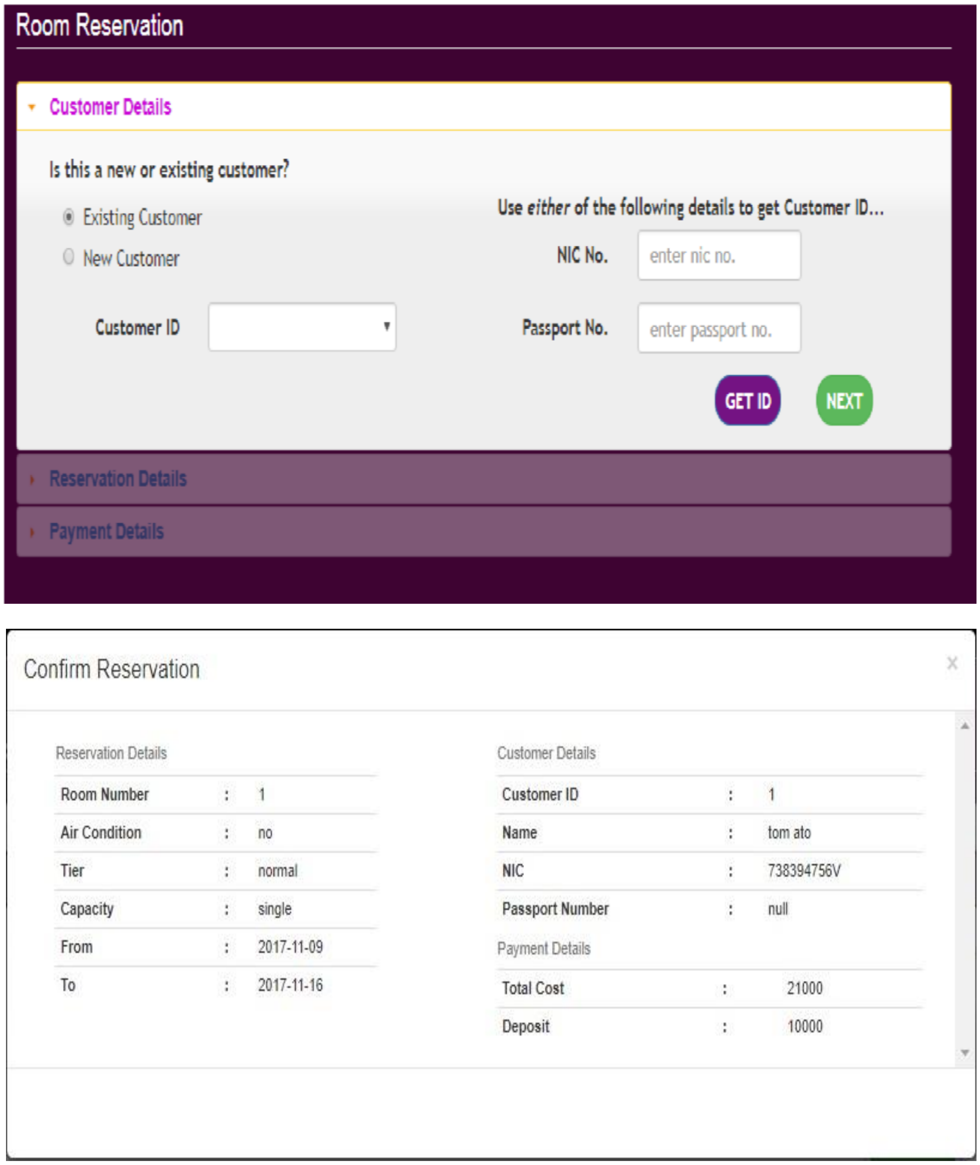
Then display payment table.

8. Add payment details.

Then system auto calculate total payment and display customer details and reservation details.

Then customer can confirm their reservation.

C.5 CUSTOMER RESERVATION FORM



Room Reservation

▾ Customer Details

Is this a new or existing customer?

● Existing Customer

○ New Customer

Customer ID

Use *either* of the following details to get Customer ID...

NIC No. enter nic no.

Passport No. enter passport no.

Confirm Reservation

Reservation Details

Room Number	:	1
Air Condition	:	no
Tier	:	normal
Capacity	:	single
From	:	2017-11-09
To	:	2017-11-16

Customer Details

Customer ID	:	1
Name	:	tom ato
NIC	:	738394756V
Passport Number	:	null

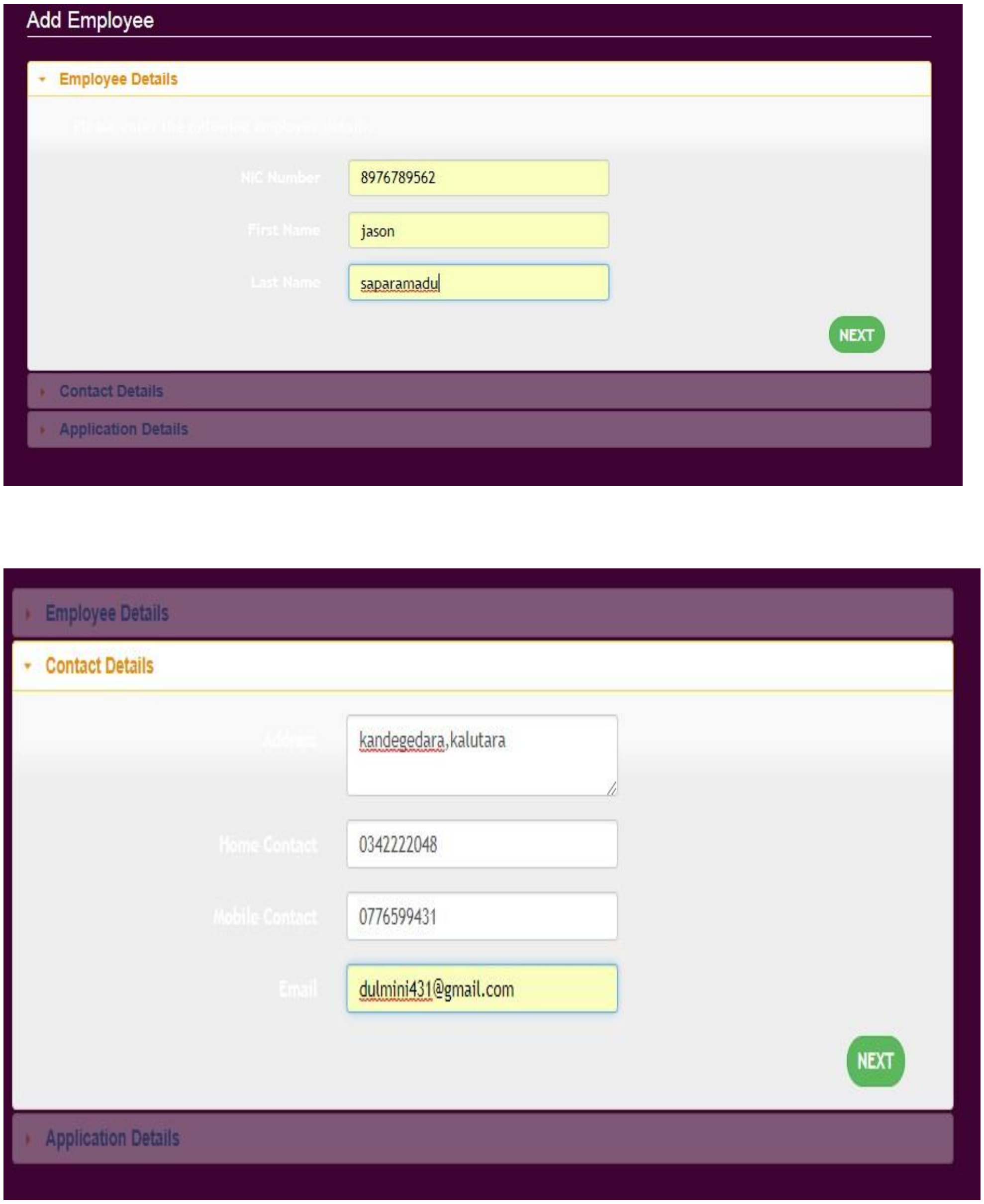
Payment Details

Total Cost	:	21000
Deposit	:	10000

Figure C.6: Customer Reservation Form

C.6 Add Employee details form

1. Enter the employee personal details
2. Click next button.
3. Display employee qualification form
4. Enter qualification details and click proceed button.



Add Employee

Please enter the following employee details.

NIC Number: 8976789562

First Name: jason

Last Name: saparamadu

Address: kandegedara,kalutara

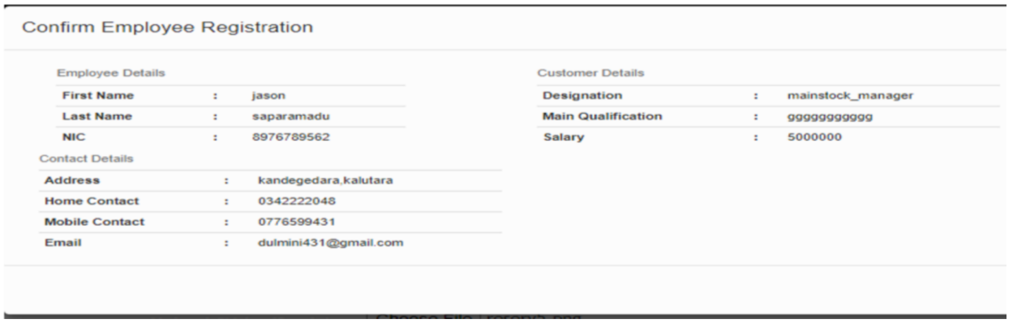
Home Contact: 0342222048

Mobile Contact: 0776599431

Email: dulmini431@gmail.com

Figure C.7: Employee Form

1. Click view button
2. Display employee detail table.



Confirm Employee Registration

Employee Details

First Name	:	jason
Last Name	:	saparamadu
NIC	:	8976789562

Contact Details

Address	:	kandegedara,kalutara
Home Contact	:	0342222048
Mobile Contact	:	0776599431
Email	:	dulmini431@gmail.com

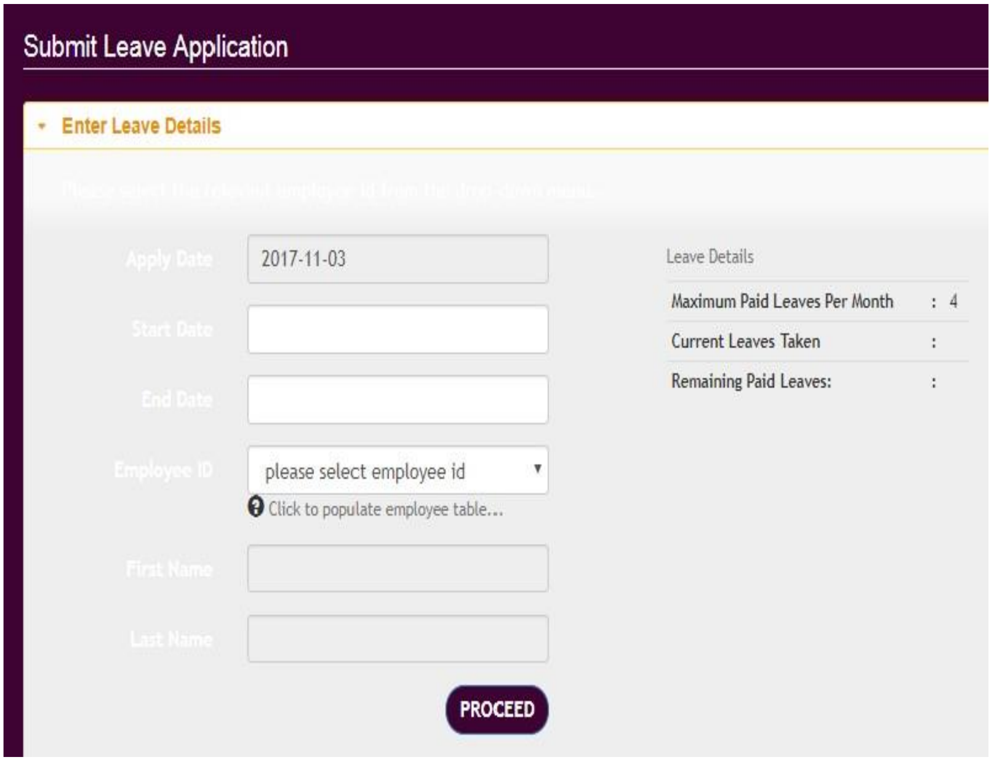
Customer Details

Designation	:	mainstock_manager
Main Qualification	:	ggggggggggg
Salary	:	5000000

Figure C.8: Employee Detail Form

C.8 Employee Leave Form.

1. Enter employee leave data.
2. Click proceed button.



Submit Leave Application

Enter Leave Details

Please select the relevant employee Id from the drop down menu.

Apply Date	2017-11-03
Start Date	
End Date	
Employee ID	please select employee id ▾
	Click to populate employee table...
First Name	
Last Name	

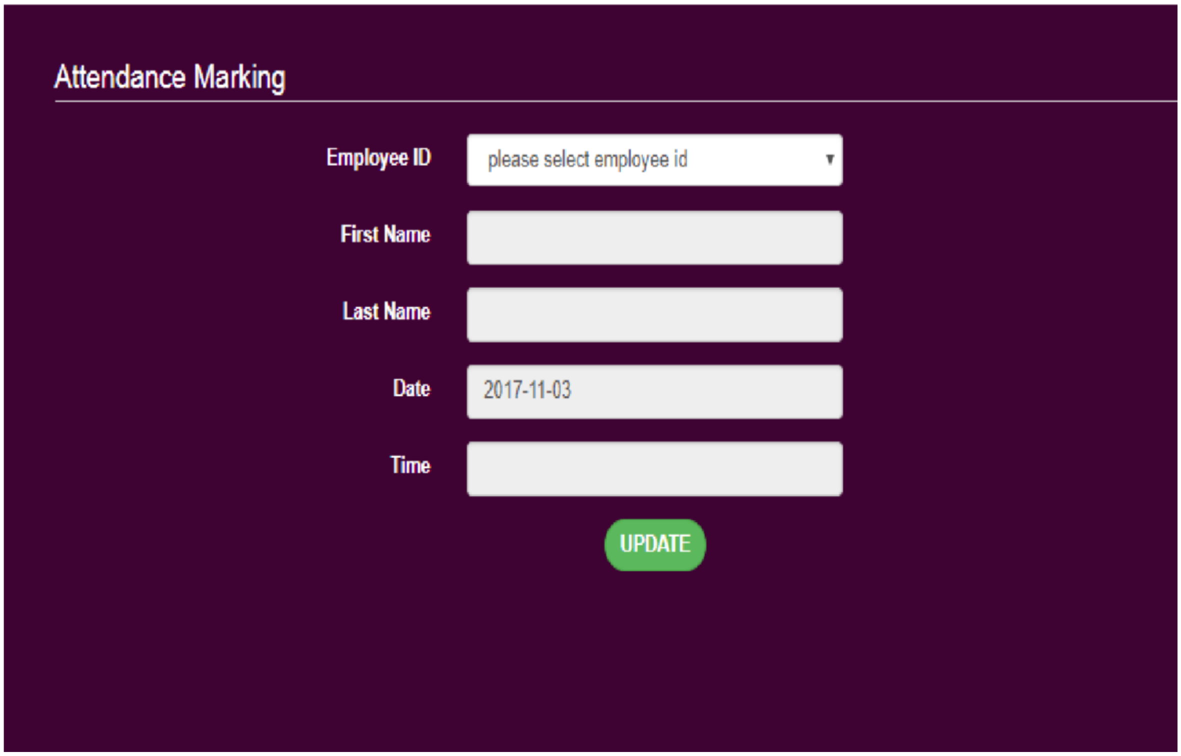
Leave Details

Maximum Paid Leaves Per Month	:	4
Current Leaves Taken	:	
Remaining Paid Leaves:	:	

1Figure C9. Employee Leave Form

C.9 Employee attendance form.

1. Enter the employee attendance data.
2. Click update button.



Attendance Marking

Employee ID: please select employee id ▾

First Name:

Last Name:

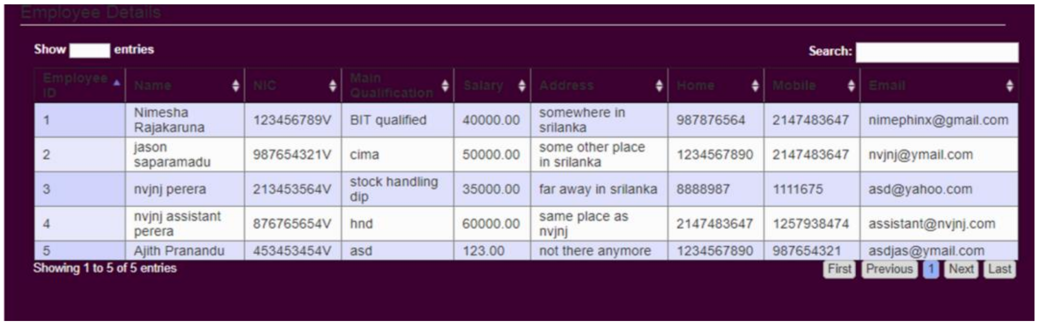
Date: 2017-11-03

Time:

Figure C.10: Employee attendance Form

C.10 Employee search form

1. Enter employee id, name or address.
2. Click search button.
3. View employee details.



Employee Details

Show [ ] entries

Search: [ ]

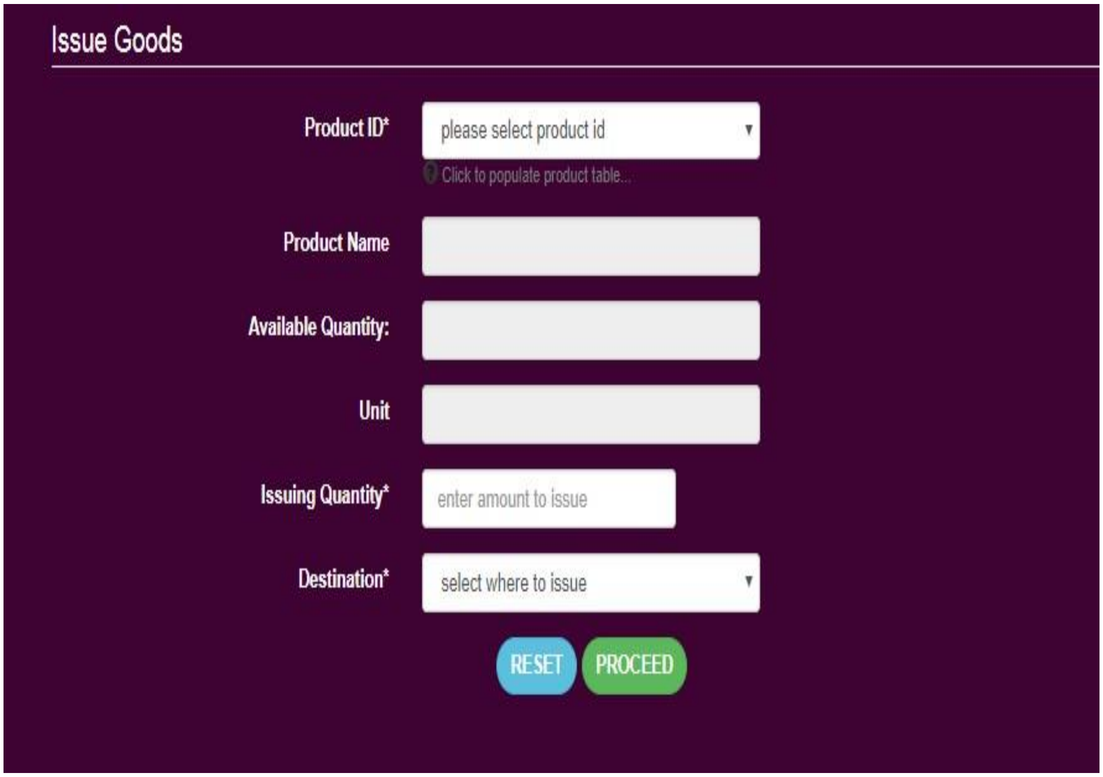
Employee ID	Name	NIC	Main Qualification	Salary	Address	Home	Mobile	Email
1	Nimesha Rajakaruna	123456789V	BIT qualified	40000.00	somewhere in srilanka	987876564	2147483647	nimephinx@gmail.com
2	jason saparamadu	987654321V	cima	50000.00	some other place in srilanka	1234567890	2147483647	nvjnj@ymail.com
3	nvjnj perera	213453564V	stock handling dip	35000.00	far away in srilanka	8888987	1111675	asd@yahoo.com
4	nvjnj assistant perera	876765654V	hnd	60000.00	same place as nvjnj	2147483647	1257938474	assistant@nvjnj.com
5	Ajith Pranandu	453453454V	asd	123.00	not there anymore	1234567890	987654321	asdjas@ymail.com

Showing 1 to 5 of 5 entries

Figure C.11 search Form

C.11 Add stock detail form.

1. Enter stock details
2. Click proceed button



Issue Goods

Product ID* please select product id ▾

Click to populate product table...

Product Name

Available Quantity:

Unit

Issuing Quantity* enter amount to issue

Destination* select where to issue ▾

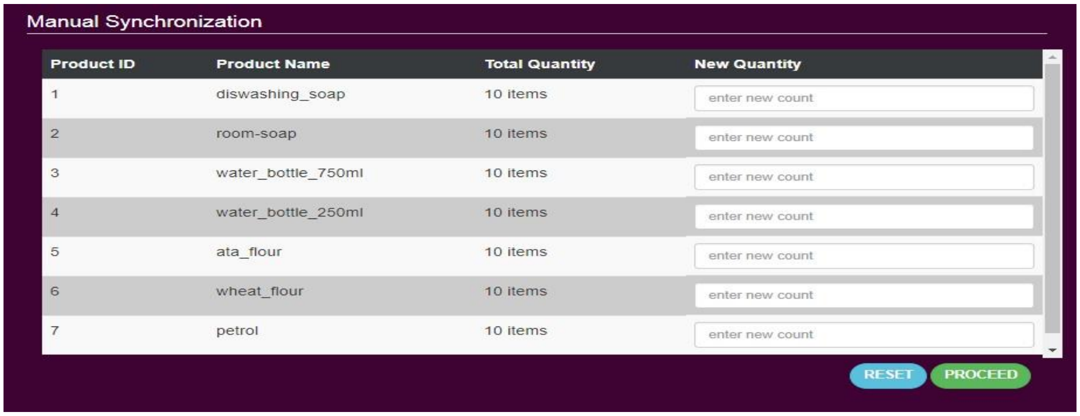
Figure C.12: stock Form

3.Click proceed button

After the click proceed button system display stock details in the table. That table showing stock type, stock quantity, receive stock quantity, receive date and auto calculate available stock quantity.

If stock available quantity less than minimum quantity, table back colon must be red.

C.12 STOCK DETAIL FORM



Manual Synchronization

Product ID	Product Name	Total Quantity	New Quantity
1	diswashing_soap	10 items	enter new count
2	room-soap	10 items	enter new count
3	water_bottle_750ml	10 items	enter new count
4	water_bottle_250ml	10 items	enter new count
5	ata_flour	10 items	enter new count
6	wheat_flour	10 items	enter new count
7	petrol	10 items	enter new count

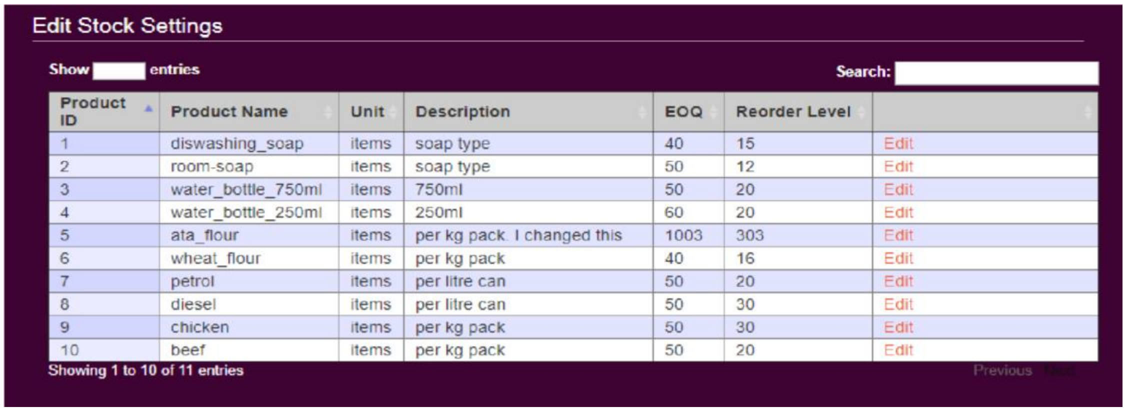
Figure C.13: Stock Detail Form

If user want to edit some details in a table

4. Click edit button.

5. Edit detail and click ok button.

C.13 STOCK EDIT FORM



Edit Stock Settings

Show entries

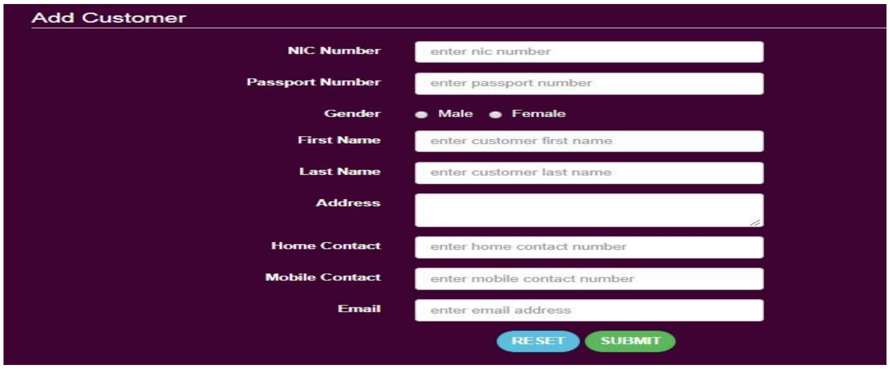
Search:

Product ID	Product Name	Unit	Description	EOQ	Reorder Level	
1	diswashing_soap	items	soap type	40	15	Edit
2	room-soap	items	soap type	50	12	Edit
3	water_bottle_750ml	items	750ml	50	20	Edit
4	water_bottle_250ml	items	250ml	60	20	Edit
5	ata_flour	items	per kg pack. I changed this	1003	303	Edit
6	wheat_flour	items	per kg pack	40	16	Edit
7	petrol	items	per litre can	50	20	Edit
8	diesel	items	per litre can	50	30	Edit
9	chicken	items	per kg pack	50	30	Edit
10	beef	items	per kg pack	50	20	Edit

Showing 1 to 10 of 11 entries

Figure C.14: Stock edit Form

C.14 Add customer details for customer form



Add Customer

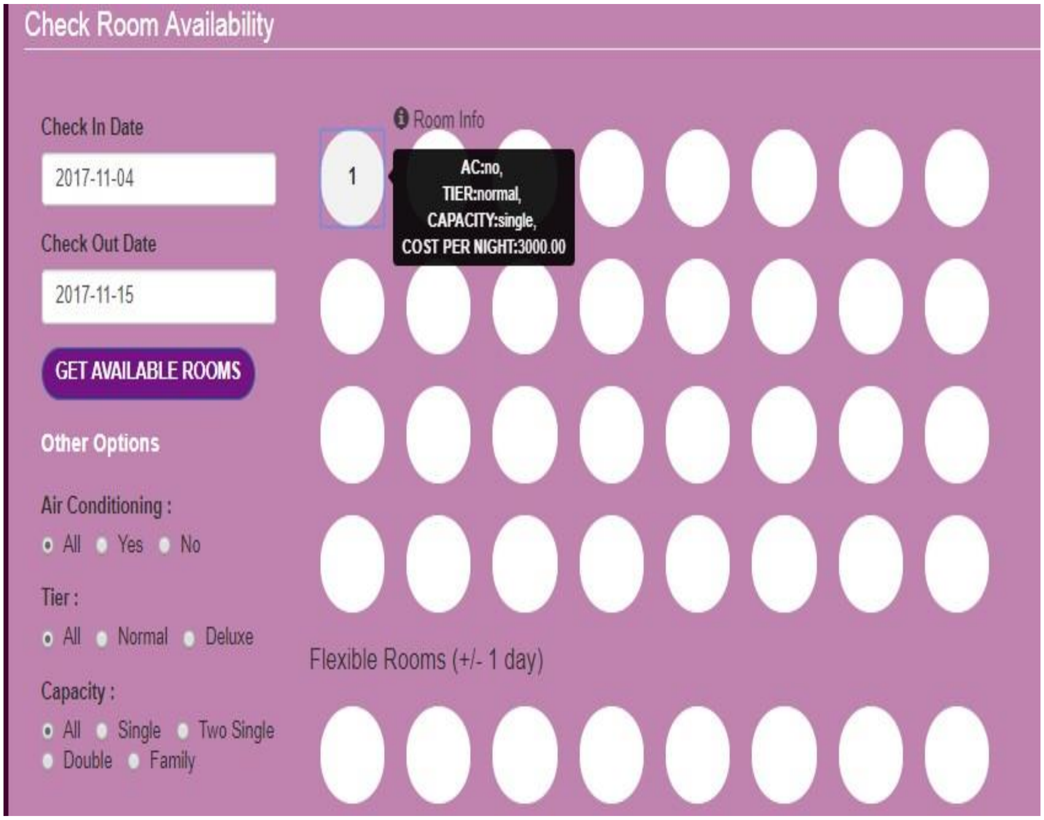
NIC Number	enter nic number
Passport Number	enter passport number
Gender	● Male ● Female
First Name	enter customer first name
Last Name	enter customer last name
Address	
Home Contact	enter home contact number
Mobile Contact	enter mobile contact number
Email	enter email address

Figure C.15: customer Form

1. Add customer details.
2. Click proceed button.

C.15 Room status form

Display room status.



Check Room Availability

Check In Date

2017-11-04

Check Out Date

2017-11-15

GET AVAILABLE ROOMS

Other Options

Air Conditioning :

● All ○ Yes ○ No

Tier :

● All ○ Normal ○ Deluxe

Capacity :

● All ○ Single ○ Two Single ○ Double ○ Family

Room Info

1

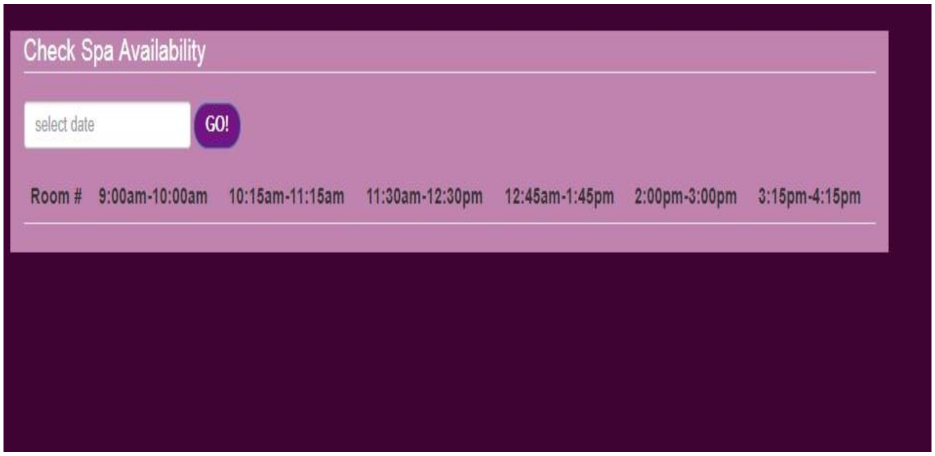
AC:no,
TIER:normal,
CAPACITY:single,
COST PER NIGHT:3000.00

Flexible Rooms (+/- 1 day)

Figure C.16: Stock status Form

C.16 Spa reservation form

System can do spa reservation.



Check Spa Availability

select date

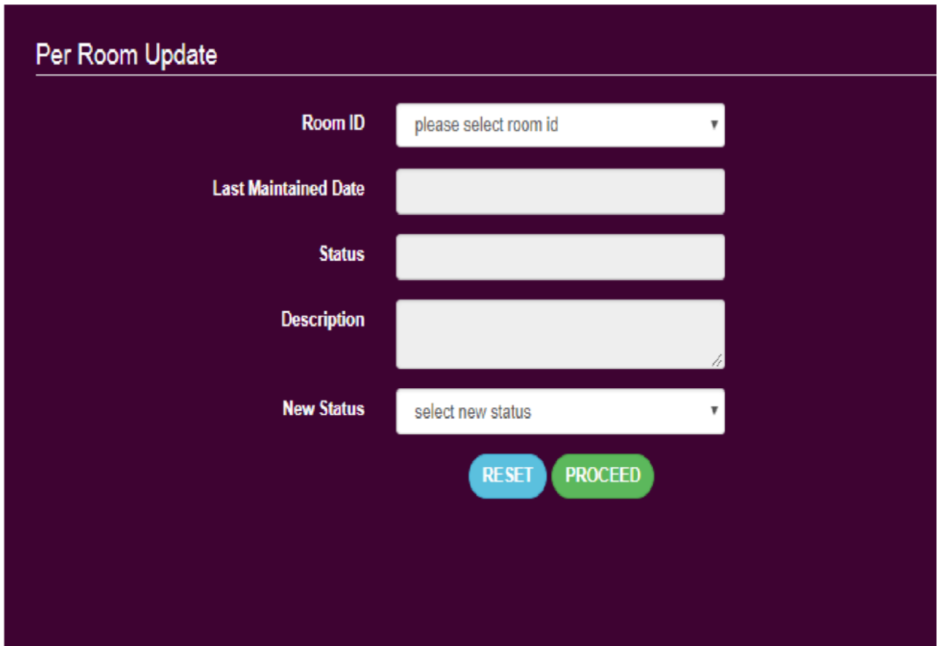
GO!

Room #	9:00am-10:00am	10:15am-11:15am	11:30am-12:30pm	12:45am-1:45pm	2:00pm-3:00pm	3:15pm-4:15pm

Figure C.17: spa reservation form

C.17 Add maintenance detail form

1. Add maintenance details.
2. Click proceed button.



Per Room Update

Room ID: please select room id ▾

Last Maintained Date:

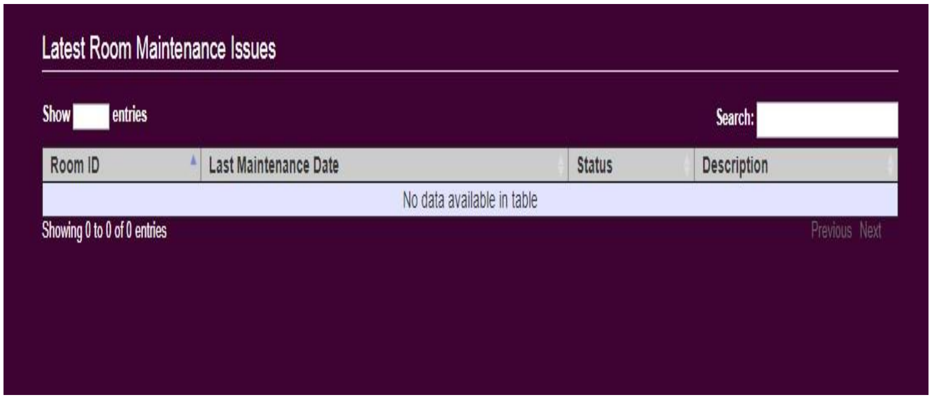
Status:

Description:

New Status: select new status ▾

Figure C.18: maintenance Form

C.18 MAINTENANCE DETAIL FORM



Latest Room Maintenance Issues

Show entries

Search:

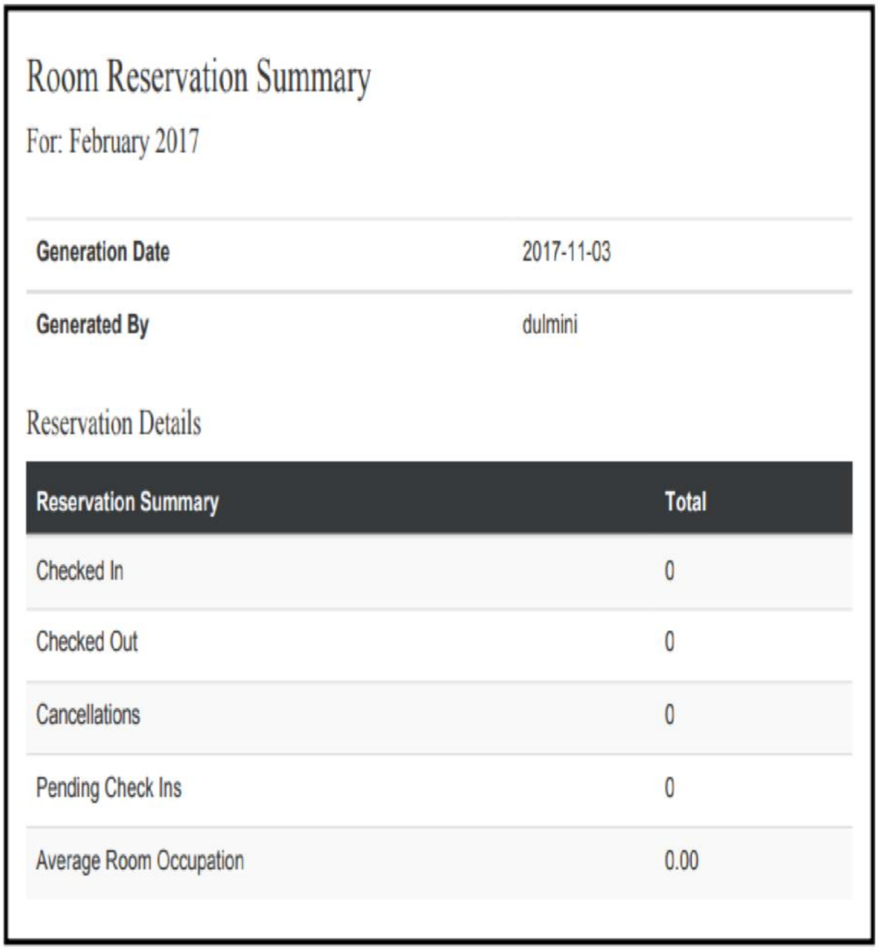
Room ID	Last Maintenance Date	Status	Description
No data available in table			

Showing 0 to 0 of 0 entries

Figure C.19: maintenance detail Form

APPENDIXD-MANAGEMENT REPORTS

D.1 ROOM RESERVATION SUMMARY PROFILE



Room Reservation Summary

For: February 2017

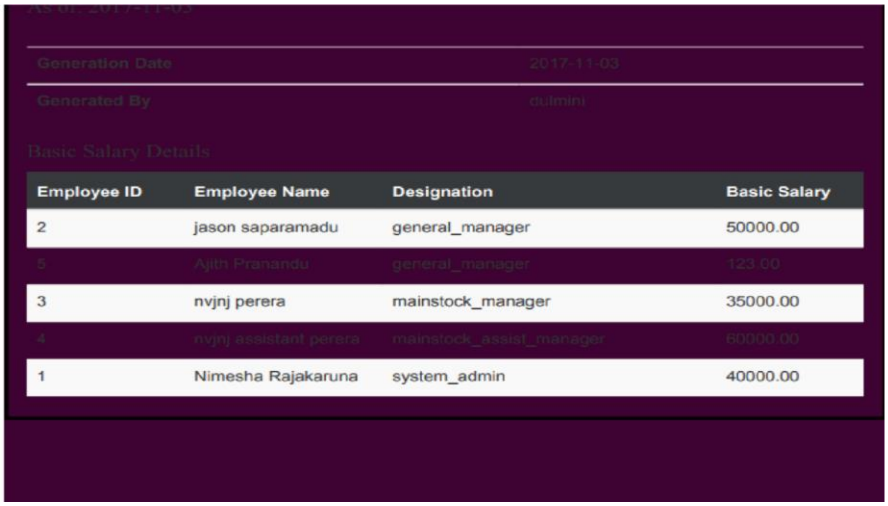
Generation Date	2017-11-03
Generated By	dulmini

Reservation Details

Reservation Summary	Total
Checked In	0
Checked Out	0
Cancellations	0
Pending Check Ins	0
Average Room Occupation	0.00

Figure D.1 room reservation summary profile

D.2 EMPLOYEE SALARY REPORT



As of: 2017-11-03

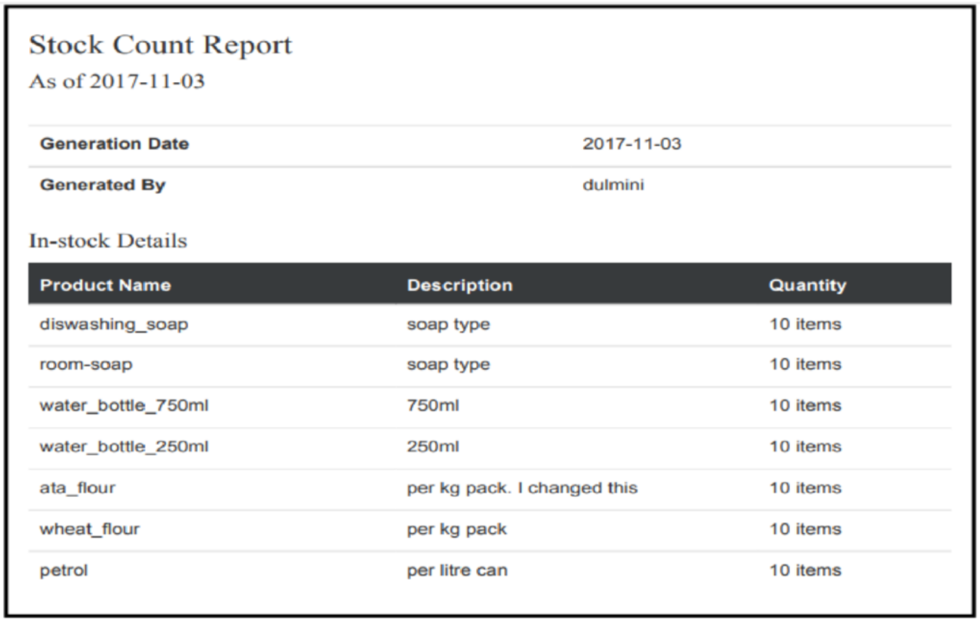
Generation Date	2017-11-03
Generated By	dulmini

Basic Salary Details

Employee ID	Employee Name	Designation	Basic Salary
2	jason saparamadu	general_manager	50000.00
5	Ajith Pranandu	general_manager	123.00
3	nvjnj perera	mainstock_manager	35000.00
4	nvjnj assistant perera	mainstock_assist_manager	60000.00
1	Nimesha Rajakaruna	system_admin	40000.00

Figure D.2 employee salary report

D.3 STOCK COUNT REPORT



Stock Count Report

As of 2017-11-03

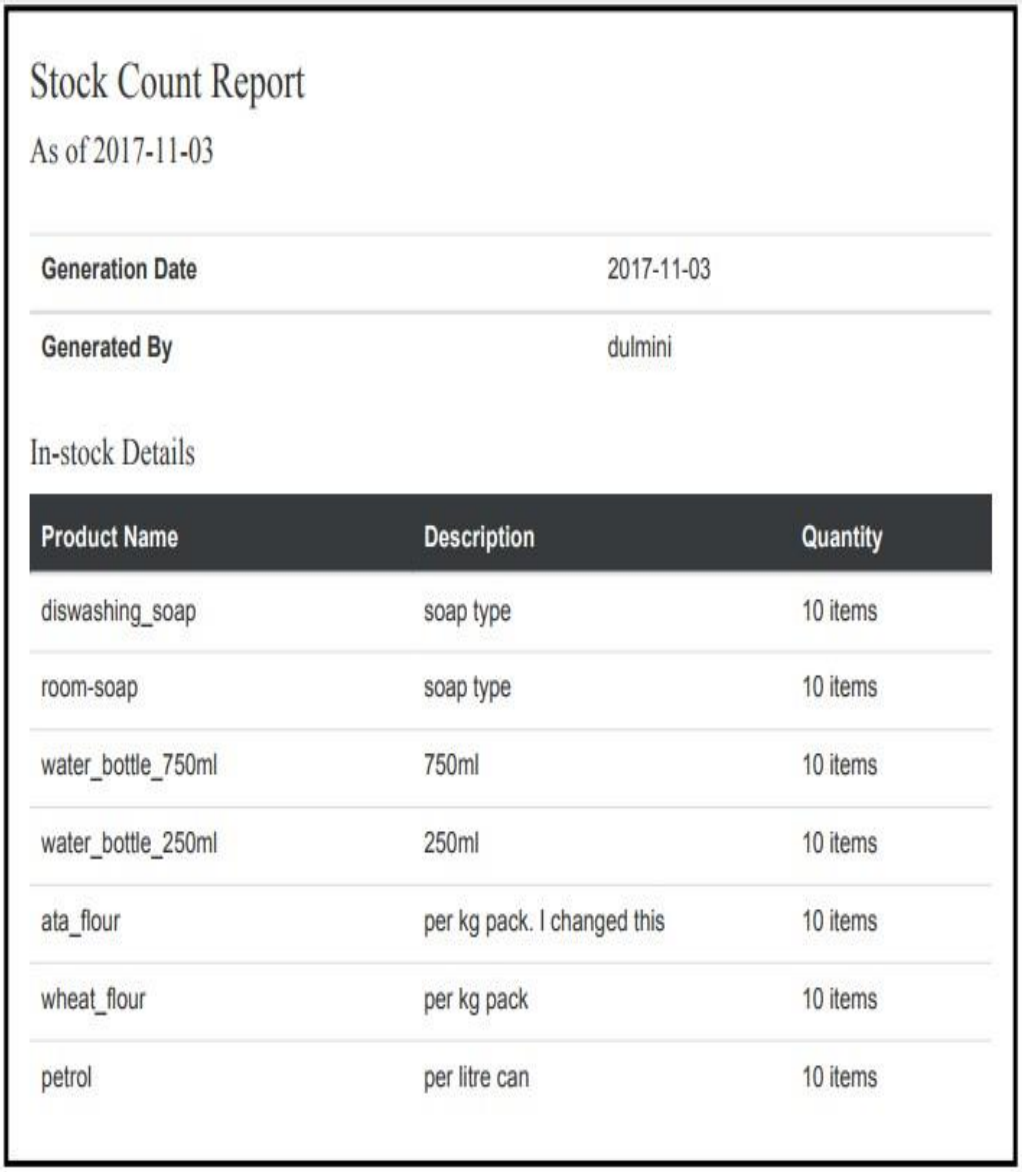
Generation Date	2017-11-03
Generated By	dulmini

In-stock Details

Product Name	Description	Quantity
diswashing_soap	soap type	10 items
room-soap	soap type	10 items
water_bottle_750ml	750ml	10 items
water_bottle_250ml	250ml	10 items
ata_flour	per kg pack. I changed this	10 items
wheat_flour	per kg pack	10 items
petrol	per litre can	10 items

Figure D.3 stock count report

D.4 STOCK STATUS REPORT



Stock Count Report

As of 2017-11-03

Generation Date	2017-11-03
Generated By	dulmini

In-stock Details

Product Name	Description	Quantity
diswashing_soap	soap type	10 items
room-soap	soap type	10 items
water_bottle_750ml	750ml	10 items
water_bottle_250ml	250ml	10 items
ata_flour	per kg pack. I changed this	10 items
wheat_flour	per kg pack	10 items
petrol	per litre can	10 items

Figure D.4 Stock status report

APENDIX E-TEST RESULT

TEST CASE RESERVATION MODULE

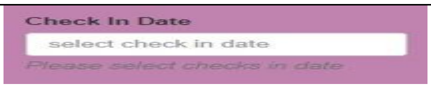
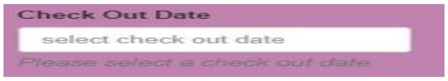
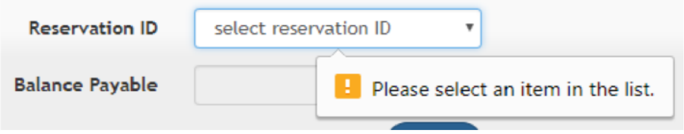
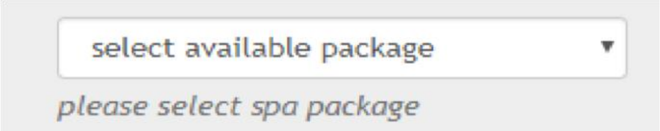
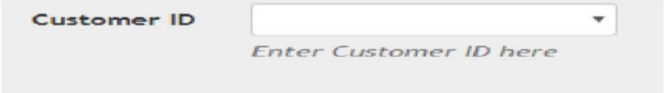
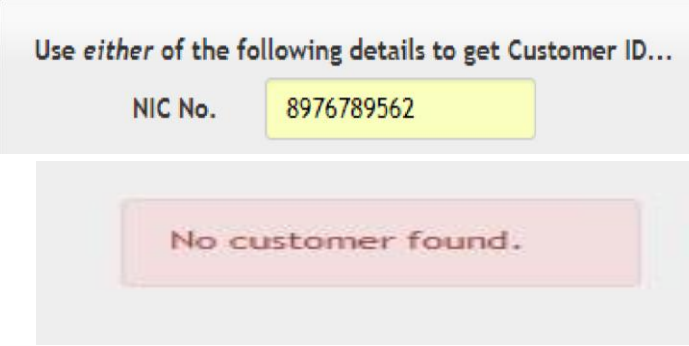
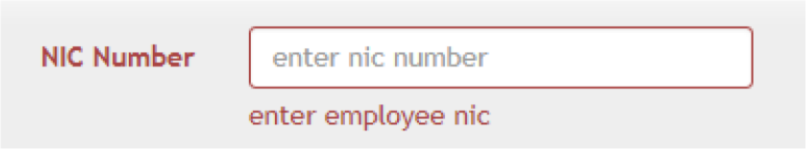
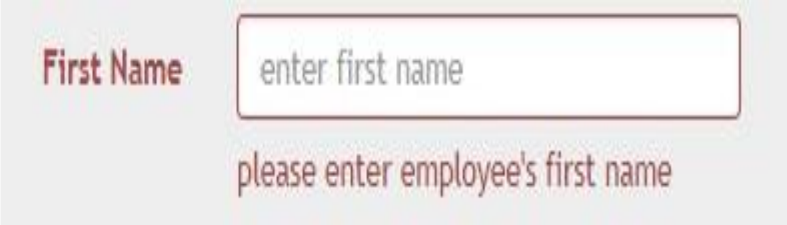
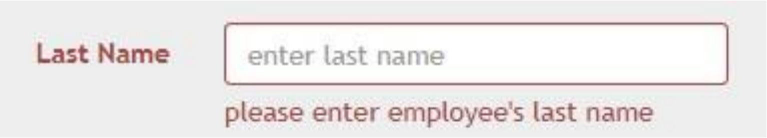
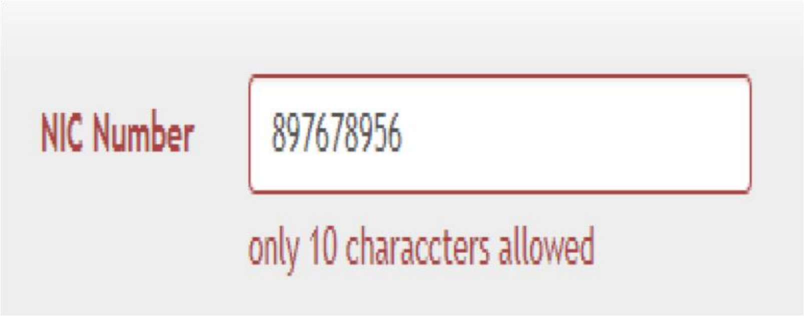
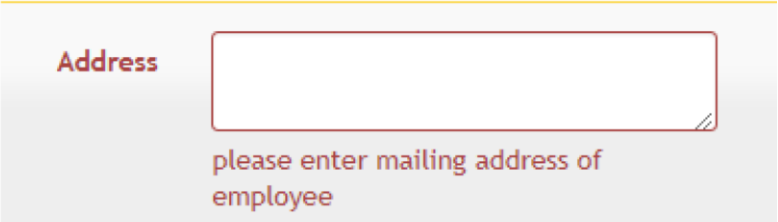
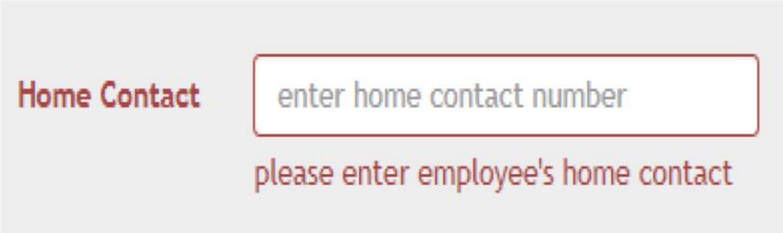
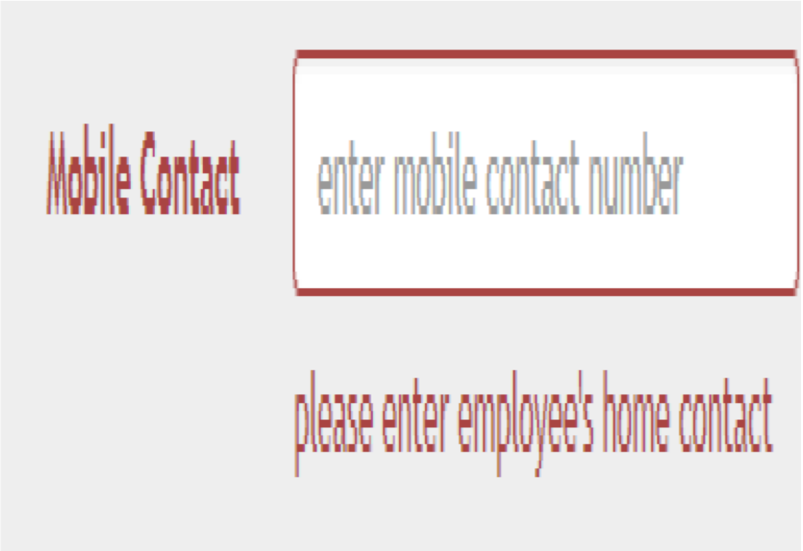
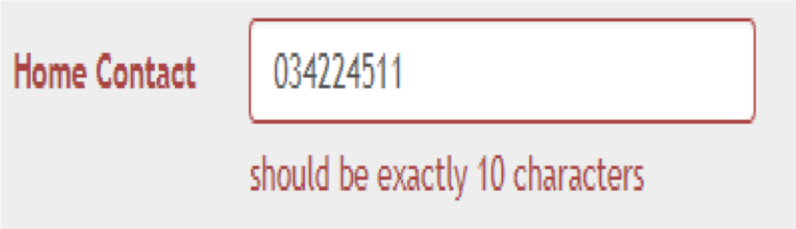
TEST CASE ID	**ACTUAL OUT PUT**	**STATUS**
1	 Check In Date select check in date Please select checks in date	pass
2	 Check Out Date select check out date Please select a check out date	pass
3	 Reservation ID: select reservation ID Balance Payable Please select an item in the list.	pass
4	 select available package please select spa package	pass
5	 Customer ID Enter Customer ID here	Pass
6	 Use *either* of the following details to get Customer ID... NIC No. 8976789562 No customer found.	Pass

Table E.1 Reservation module test result

TEST CASE EMPLOYEE MODULE

8	 NIC Number enter nic number enter employee nic	pass
9	 First Name enter first name please enter employee's first name	pass
10	 Last Name enter last name please enter employee's last name	pass
11	 NIC Number 897678956 only 10 characcters allowed	pass
12	 Address please enter mailing address of employee	pass

13	 Home Contact enter home contact number please enter employee's home contact	Pass
14	 Mobile Contact enter mobile contact number please enter employee's home contact	Pass
15	 Home Contact 034224511 should be exactly 10 characters	pass

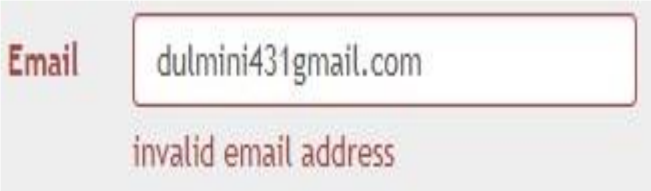
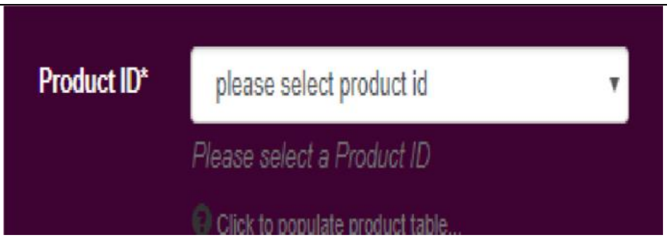
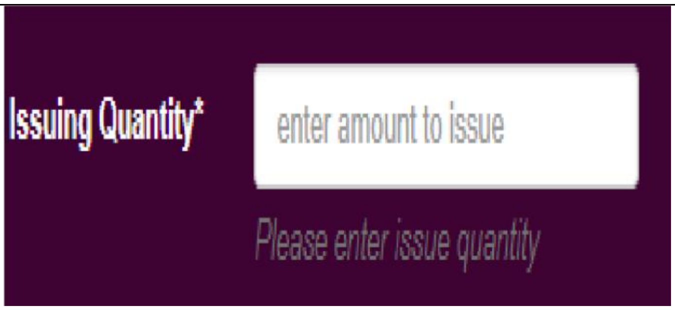
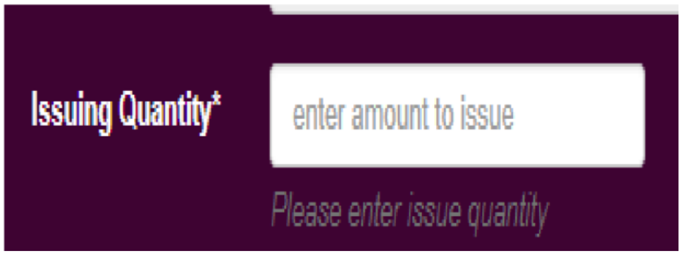
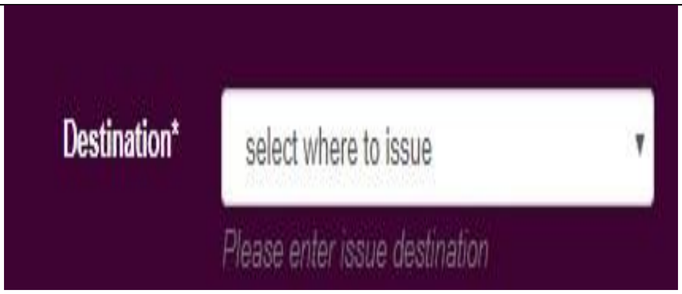
16	 Email dulmini431gmail.com invalid email address	pass

Table E.2 employee module test result

TEST CASE STOCK MANAGEMENT MODULE

17	 Product ID* please select product id ▾ Please select a Product ID Click to populate product table...	pass
18	 Issuing Quantity* enter amount to issue Please enter issue quantity	Pass
19	 Issuing Quantity* enter amount to issue Please enter issue quantity	pass

20	 Destination* select where to issue Please enter issue destination	

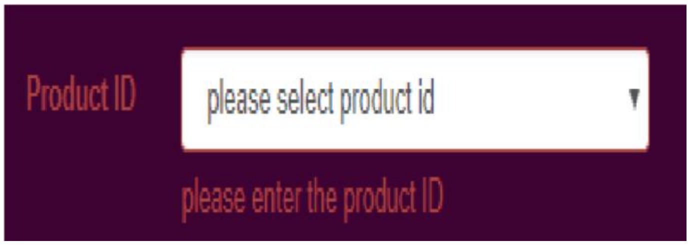
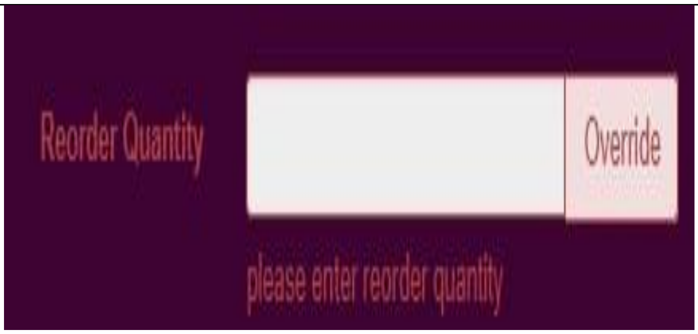
21	 Product ID please select product id please enter the product ID	pass
22	 Reorder Quantity Override please enter reorder quantity	pass

Table E.4 stock module test result

LOGIN MANAGEMENT MODULE

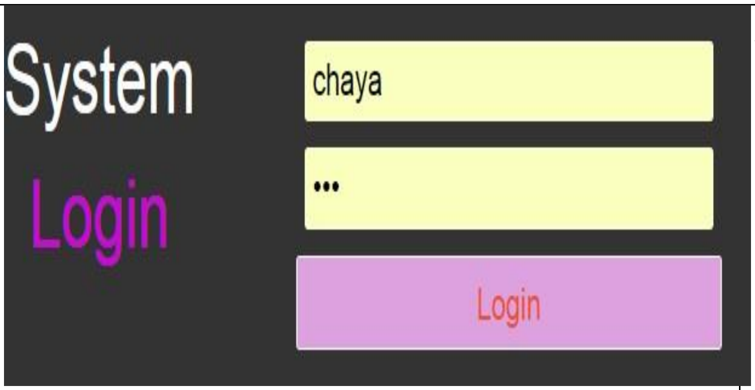
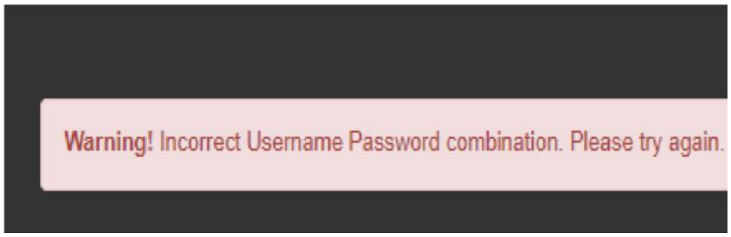
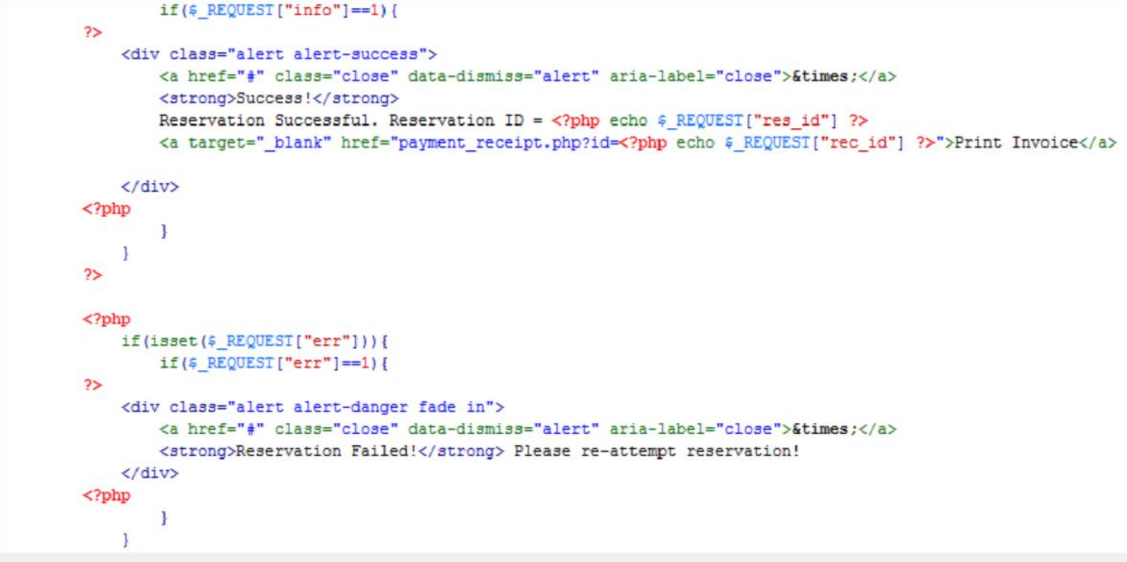
23	 chaya •••	pass
24	 **Warning!** Incorrect Username Password combination. Please try again.	pass

Table E.5 logging module test result

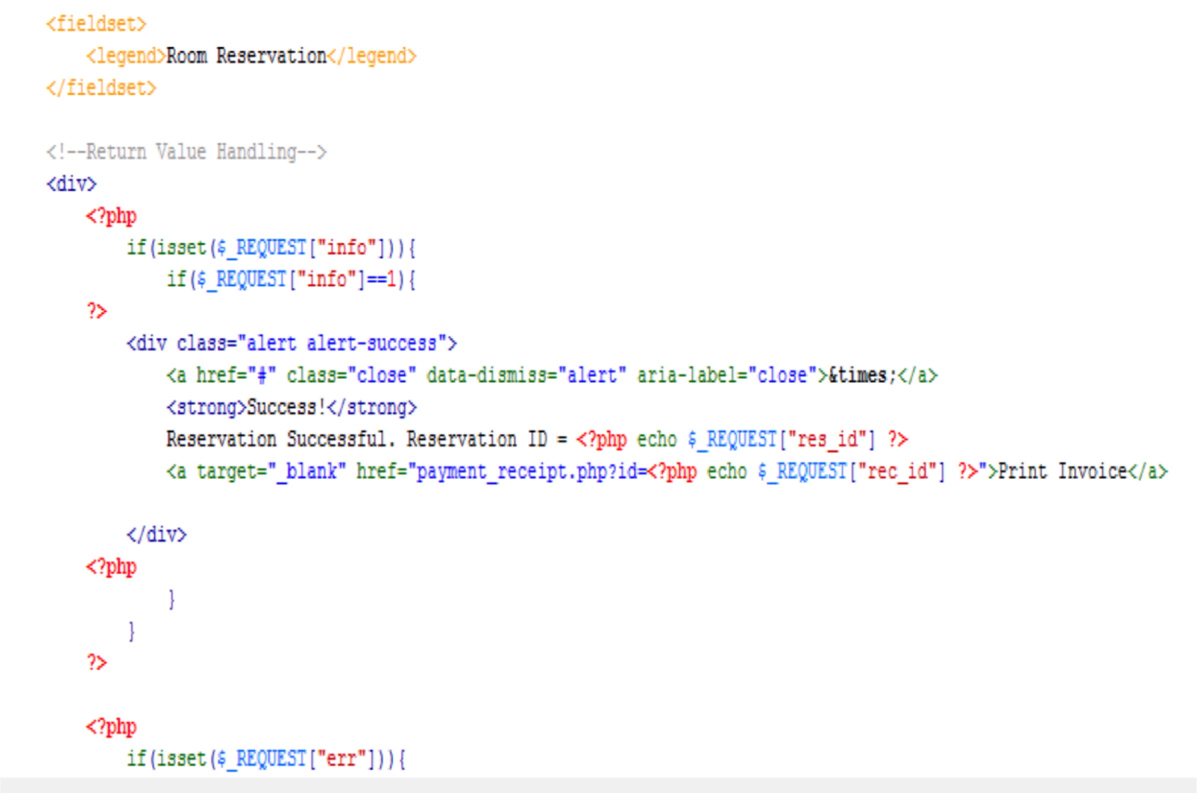
APPENDIX F-CODE LISTING

- Code for room details form

Note that all object references were not shown because of space limit

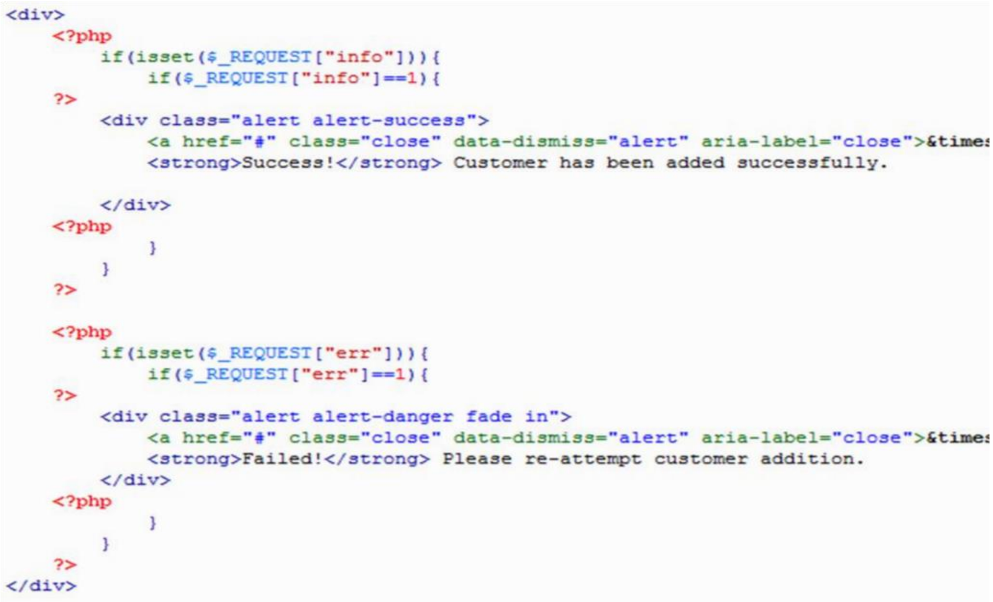


```
            if($_REQUEST["info"]==1){
    ?>
        <div class="alert alert-success">
            <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
            <strong>Success!</strong>
            Reservation Successful. Reservation ID = <?php echo $_REQUEST["res_id"] ?>
            <a target="_blank" href="payment_receipt.php?id=<?php echo $_REQUEST["rec_id"] ?>">Print Invoice</a>

        </div>
    <?php
            }
        }
    ?>

    <?php
        if(isset($_REQUEST["err"])){
            if($_REQUEST["err"]==1){
    ?>
        <div class="alert alert-danger fade in">
            <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
            <strong>Reservation Failed!</strong> Please re-attempt reservation!
        </div>
    <?php
            }
        }
```

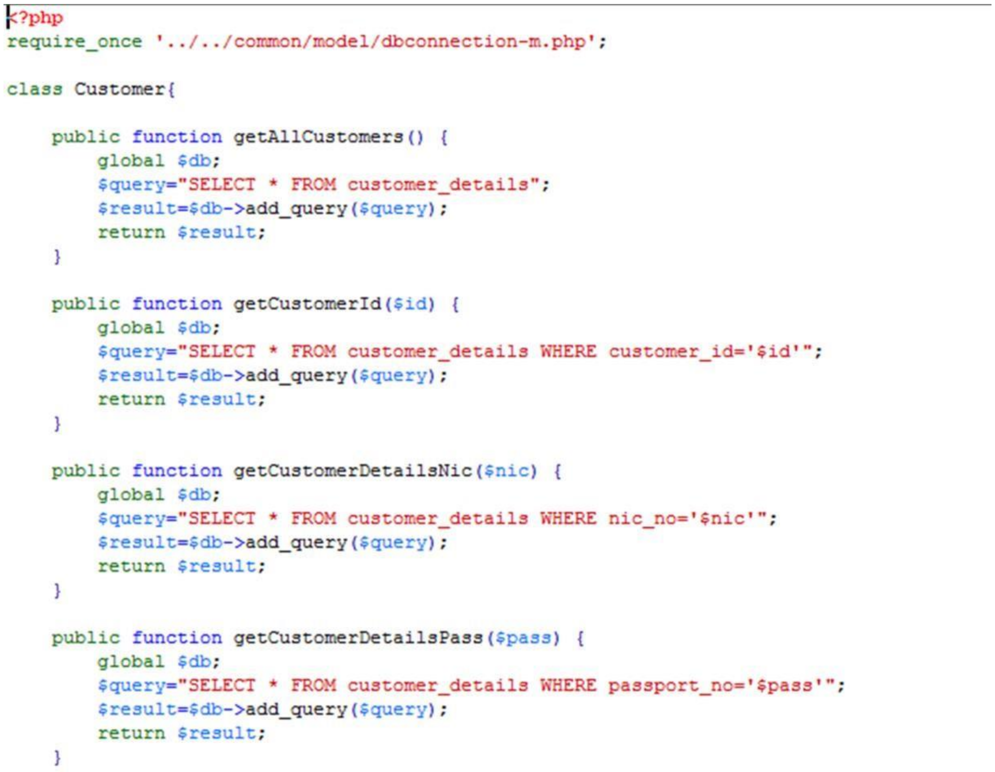


```
<fieldset>
    <legend>Room Reservation</legend>
</fieldset>

<!--Return Value Handling-->
<div>
    <?php
        if(isset($_REQUEST["info"])){
            if($_REQUEST["info"]==1){
    ?>
        <div class="alert alert-success">
            <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
            <strong>Success!</strong>
            Reservation Successful. Reservation ID = <?php echo $_REQUEST["res_id"] ?>
            <a target="_blank" href="payment_receipt.php?id=<?php echo $_REQUEST["rec_id"] ?>">Print Invoice</a>

        </div>
    <?php
            }
        }
    ?>

    <?php
        if(isset($_REQUEST["err"])){
```

- Code for add customer details



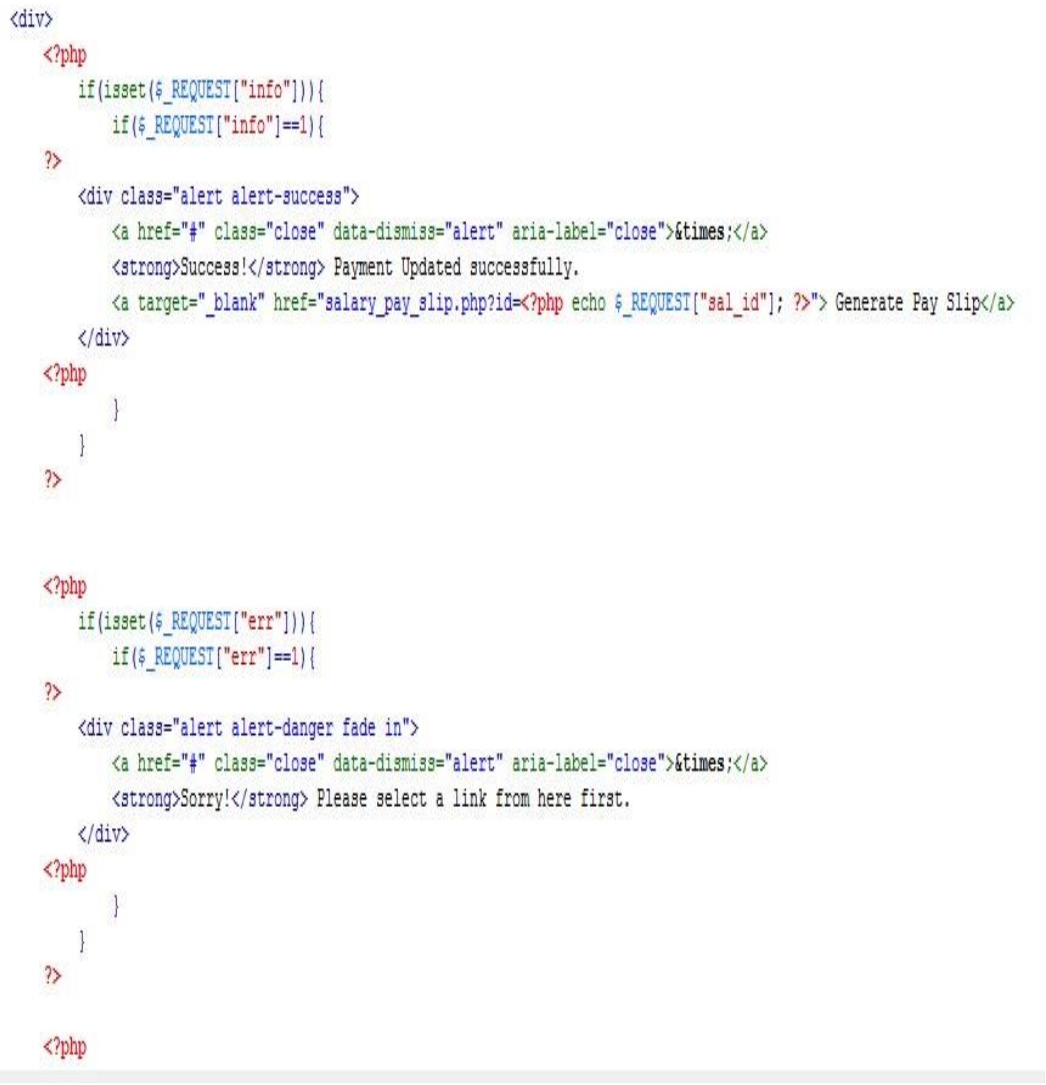
```
<div>
    <?php
        if(isset($_REQUEST["info"])){
            if($_REQUEST["info"]==1){
    ?>
        <div class="alert alert-success">
            <a href="#" class="close" data-dismiss="alert" aria-label="close">&times
            <strong>Success!</strong> Customer has been added successfully.

        </div>
    <?php
            }
        }
    ?>

    <?php
        if(isset($_REQUEST["err"])){
            if($_REQUEST["err"]==1){
    ?>
        <div class="alert alert-danger fade in">
            <a href="#" class="close" data-dismiss="alert" aria-label="close">&times
            <strong>Failed!</strong> Please re-attempt customer addition.
        </div>
    <?php
            }
        }
    ?>
</div>
```



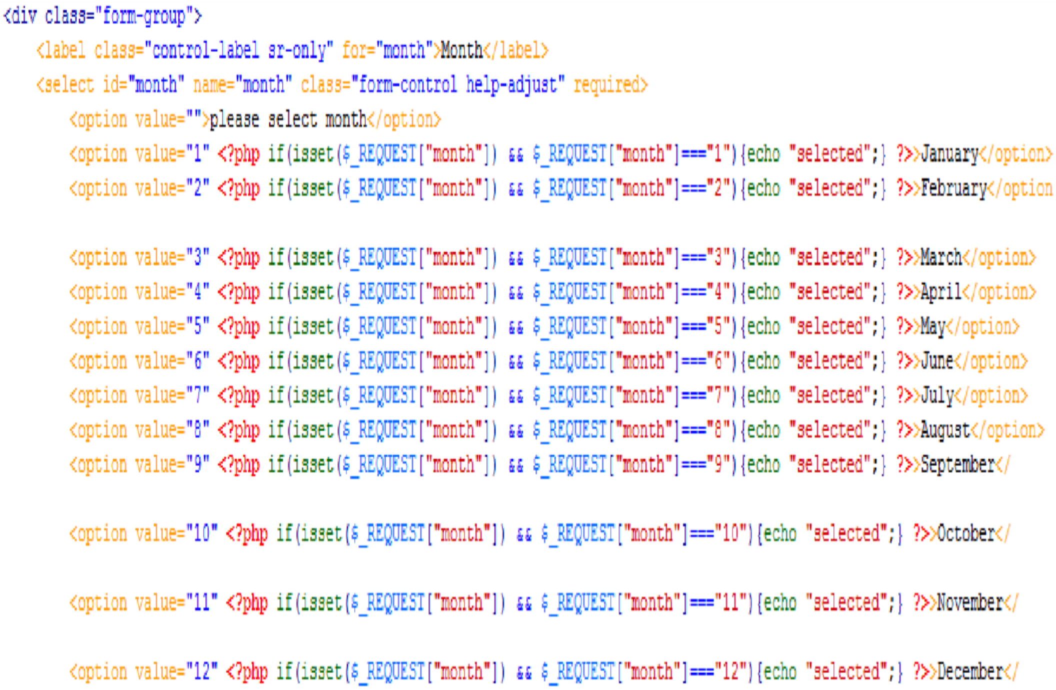
```
<?php
require_once '../../common/model/dbconnection-m.php';

class Customer{

    public function getAllCustomers() {
        global $db;
        $query="SELECT * FROM customer_details";
        $result=$db->add_query($query);
        return $result;
    }

    public function getCustomerId($id) {
        global $db;
        $query="SELECT * FROM customer_details WHERE customer_id='$id'";
        $result=$db->add_query($query);
        return $result;
    }

    public function getCustomerDetailsNic($nic) {
        global $db;
        $query="SELECT * FROM customer_details WHERE nic_no='$nic'";
        $result=$db->add_query($query);
        return $result;
    }

    public function getCustomerDetailsPass($pass) {
        global $db;
        $query="SELECT * FROM customer_details WHERE passport_no='$pass'";
        $result=$db->add_query($query);
        return $result;
    }
```

- Code for pay sheet

Note that all object references were not shown because of space limit.

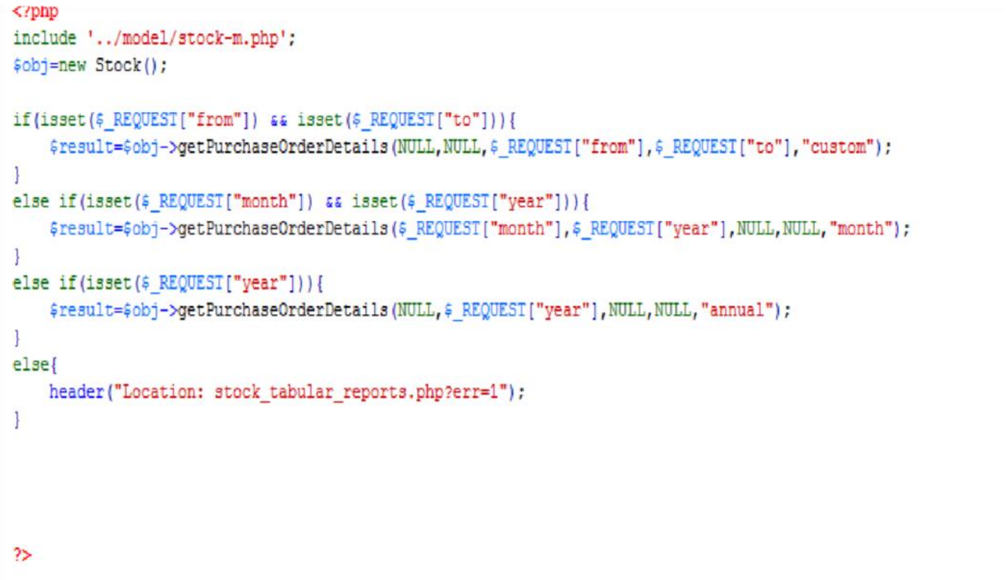


```
<div>
    <?php
        if(isset($_REQUEST["info"])){
            if($_REQUEST["info"]==1){
    ?>
        <div class="alert alert-success">
            <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
            <strong>Success!</strong> Payment Updated successfully.
            <a target="_blank" href="salary_pay_slip.php?id=<?php echo $_REQUEST["sal_id"]; ?>"> Generate Pay Slip</a>
        </div>
    <?php
            }
        }
    ?>


    <?php
        if(isset($_REQUEST["err"])){
            if($_REQUEST["err"]==1){
    ?>
        <div class="alert alert-danger fade in">
            <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
            <strong>Sorry!</strong> Please select a link from here first.
        </div>
    <?php
            }
        }
    ?>

    <?php
```

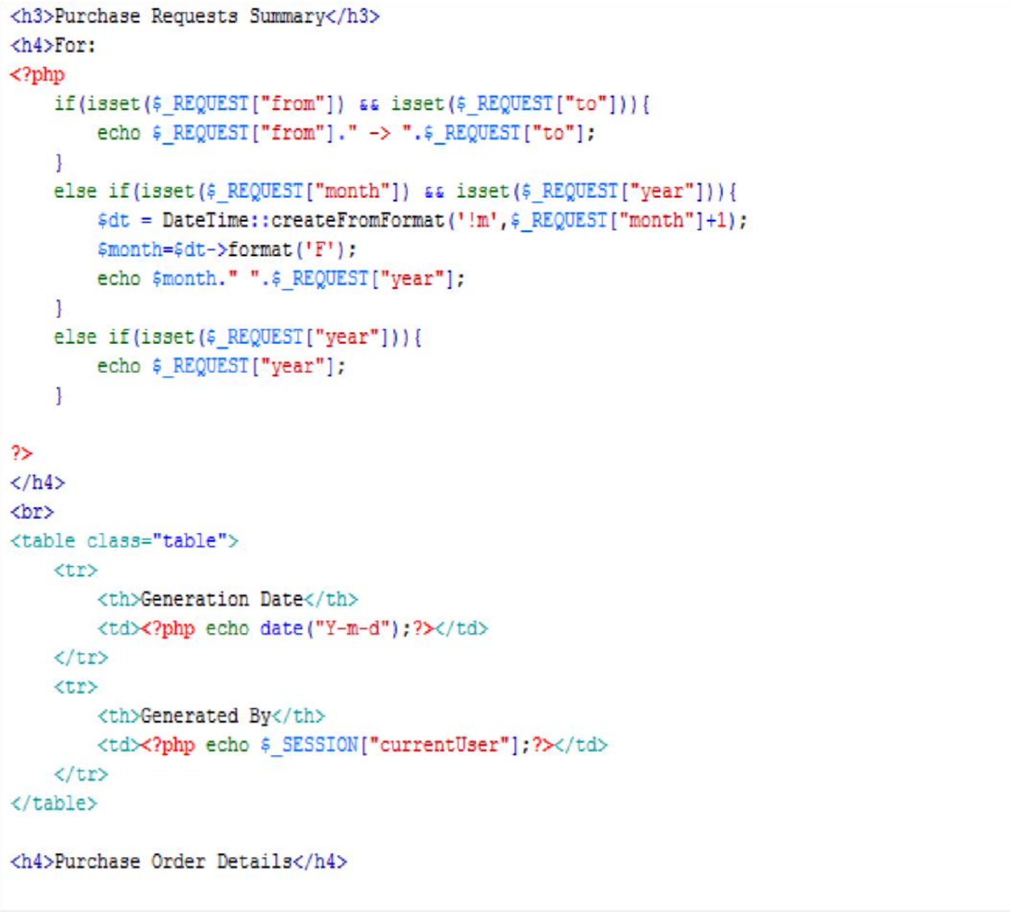


```
<div class="form-group">
    <label class="control-label sr-only" for="month">Month</label>
    <select id="month" name="month" class="form-control help-adjust" required>
        <option value="">please select month</option>
        <option value="1" <?php if(isset($_REQUEST["month"]) && $_REQUEST["month"]==="1"){echo "selected";} ?>>January</option>
        <option value="2" <?php if(isset($_REQUEST["month"]) && $_REQUEST["month"]==="2"){echo "selected";} ?>>February</option

        <option value="3" <?php if(isset($_REQUEST["month"]) && $_REQUEST["month"]==="3"){echo "selected";} ?>>March</option>
        <option value="4" <?php if(isset($_REQUEST["month"]) && $_REQUEST["month"]==="4"){echo "selected";} ?>>April</option>
        <option value="5" <?php if(isset($_REQUEST["month"]) && $_REQUEST["month"]==="5"){echo "selected";} ?>>May</option>
        <option value="6" <?php if(isset($_REQUEST["month"]) && $_REQUEST["month"]==="6"){echo "selected";} ?>>June</option>
        <option value="7" <?php if(isset($_REQUEST["month"]) && $_REQUEST["month"]==="7"){echo "selected";} ?>>July</option>
        <option value="8" <?php if(isset($_REQUEST["month"]) && $_REQUEST["month"]==="8"){echo "selected";} ?>>August</option>
        <option value="9" <?php if(isset($_REQUEST["month"]) && $_REQUEST["month"]==="9"){echo "selected";} ?>>September</

        <option value="10" <?php if(isset($_REQUEST["month"]) && $_REQUEST["month"]==="10"){echo "selected";} ?>>October</

        <option value="11" <?php if(isset($_REQUEST["month"]) && $_REQUEST["month"]==="11"){echo "selected";} ?>>November</

        <option value="12" <?php if(isset($_REQUEST["month"]) && $_REQUEST["month"]==="12"){echo "selected";} ?>>December</
```

- Code for stock details form.

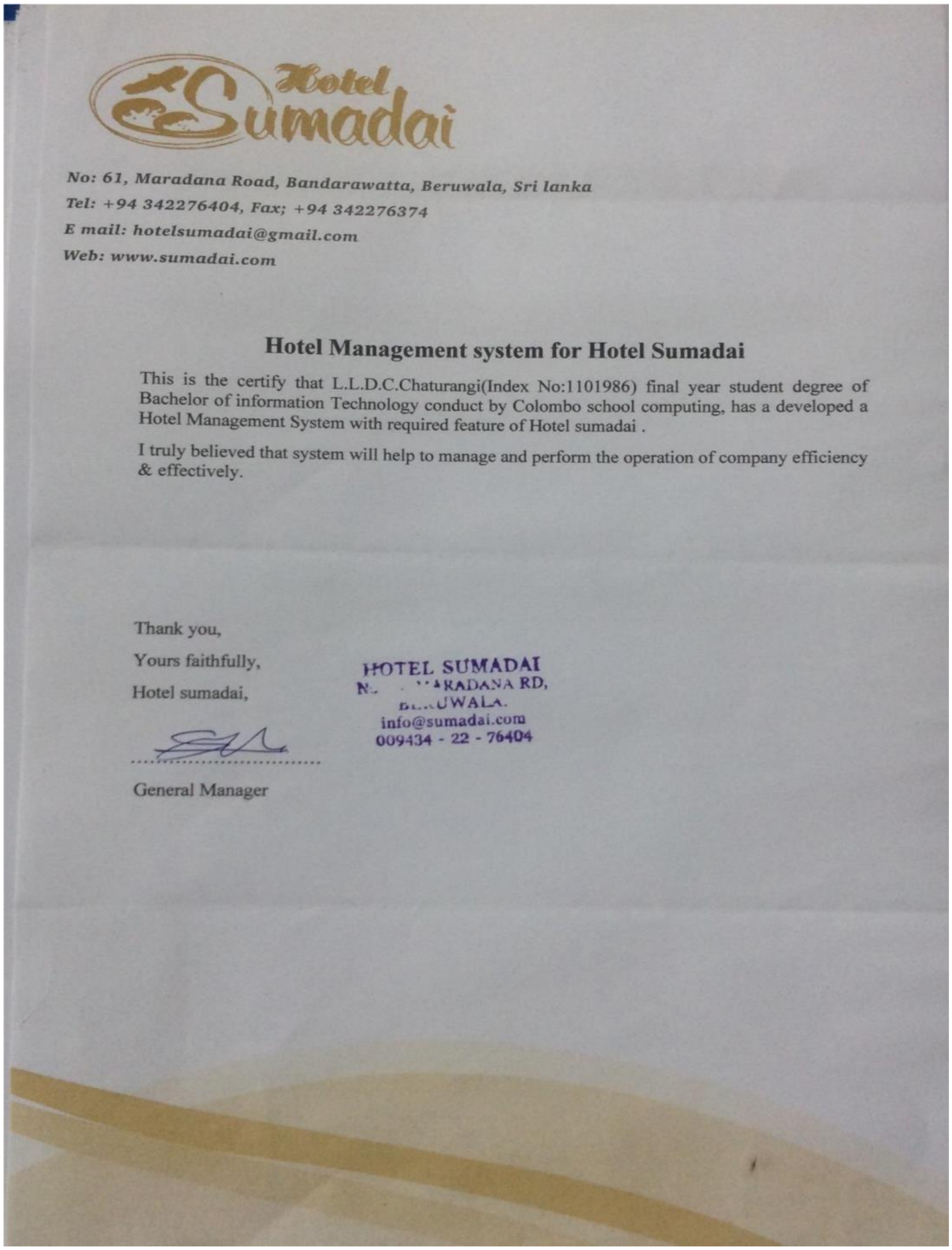


```
<?php
include '../model/stock-m.php';
$obj=new Stock();

if(isset($_REQUEST["from"]) && isset($_REQUEST["to"])){
    $result=$obj->getPurchaseOrderDetails(NULL,NULL,$_REQUEST["from"],$_REQUEST["to"],"custom");
}
else if(isset($_REQUEST["month"]) && isset($_REQUEST["year"])){
    $result=$obj->getPurchaseOrderDetails($_REQUEST["month"],$_REQUEST["year"],NULL,NULL,"month");
}
else if(isset($_REQUEST["year"])){
    $result=$obj->getPurchaseOrderDetails(NULL,$_REQUEST["year"],NULL,NULL,"annual");
}
else{
    header("Location: stock_tabular_reports.php?err=1");
}




?>
```



```
<h3>Purchase Requests Summary</h3>
<h4>For:
<?php
    if(isset($_REQUEST["from"]) && isset($_REQUEST["to"])){
        echo $_REQUEST["from"]." -> ".$_REQUEST["to"];
    }
    else if(isset($_REQUEST["month"]) && isset($_REQUEST["year"])){
        $dt = DateTime::createFromFormat('!m',$_REQUEST["month"]+1);
        $month=$dt->format('F');
        echo $month." ".$_REQUEST["year"];
    }
    else if(isset($_REQUEST["year"])){
        echo $_REQUEST["year"];
    }

?>
</h4>
<br>
<table class="table">
    <tr>
        <th>Generation Date</th>
        <td><?php echo date("Y-m-d");?></td>
    </tr>
    <tr>
        <th>Generated By</th>
        <td><?php echo $_SESSION["currentUser"];?></td>
    </tr>
</table>

<h4>Purchase Order Details</h4>
```

APPENDIXG-CLIENT CERTIFICATE



Hotel Sumadai

No: 61, Maradana Road, Bandarawatta, Beruwala, Sri lanka
Tel: +94 342276404, Fax; +94 342276374
E mail: hotelsumadai@gmail.com
Web: www.sumadai.com

Hotel Management system for Hotel Sumadai

This is the certify that L.L.D.C.Chaturangi(Index No:1101986) final year student degree of Bachelor of information Technology conduct by Colombo school computing, has a developed a Hotel Management System with required feature of Hotel sumadai .

I truly believed that system will help to manage and perform the operation of company efficiency & effectively.

Thank you,
Yours faithfully,
Hotel sumadai,

....
General Manager

HOTEL SUMADAI
N.. [illegible] ARADANA RD,
[illegible]UWALA.
info@sumadai.com
009434 - 22 - 76404

GLOSSARY

Term	Description
Use Case Diagram	Diagram which shows the overall picture
Class Diagram	Class diagram shows the relationship between system classes
Sequence Diagram	A sequence diagram shows object interactions arranged in time sequence.
Activity Diagram	An activity diagram shows the overall flow of control
Rational Unified Process	Iterative software development process

INDEX