



# **Online Tour Package Reservation System**

**A dissertation submitted for the Degree of Master of  
Information Technology**

**R.P.U.L. Rajapaksha**

**University of Colombo School of Computing**

**2019**



## Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name : **R.P. Udula Laksitha Rajapaksha**

Registration Number : **2016/MIT/034**

Index Number : **16550345**

---

Signature:

Date:

This is to certify that this thesis is based on the work of Mr. R.P. Udula Laksitha Rajapaksha under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name : **Dr. K.L. Jayaratne**

---

Signature:

Date:

## **Abstract**

This Tour Package Reservation is a web based system. There are 3 type of users in this system. They are administrator, Customer and guest. The system allows the user to easily manage the shopping process where an online cart will allow the customer to buy from our outlet from their home. The shopping process such as browsing for product and manual payment method using cash which customer nowadays use is time consuming. The development of this system is done by using PHP language and MySQL as the database management.

Web applications are popular due to the ubiquity of the browser as a client, sometimes called thin client. The ability to update and maintain web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity. System will present a mostly complete view of application focusing especially on the design process. The solution was developed as an online web application. Client will use a web browser to access the application.

Finally the system fully tested with most common browsers and an evaluation is also done using a survey and the outcomes discussed.

## **Acknowledgement**

I would like to thank the Directors and all the academic staff of University of Colombo School of Computing for providing me the opportunity to follow the Master of Information Technology Degree programme and all the lecturers who gave me the knowledge in the course duration.

My sincere gratitude to Dr. K.L. Jayaratne, the supervisor of the project for the tremendous guidance, motivation and cooperation given throughout the year.

My special thanks go to the society who helped me in filling out the questionnaire which I have distributed to gather requirements.

Also I thank all my friends who helped me a lot in different ways and encouraging me to finalise the project successfully.

Thank you !

## Table of Contents

Declaration.....	ii
Abstract .....	iii
Acknowledgement .....	iv
Table of Contents .....	v
List of Figures .....	viii
List of Tables .....	x
List of Abbreviations .....	xi

### CHAPTER 1

INTRODUCTION.....	1
1.1 Motivation .....	1
1.2 Problem Definition .....	2
1.3 Objectives .....	3
1.4 Project Significance .....	3
1.5 Project Scope .....	4
1.6 Flow of thesis .....	5

### CHAPTER 2

REQUIREMENT ANALYSISAND RELATED WORKS .....	7
2.1 Requirement Analysis .....	7
2.2 Reservation System .....	8
2.3 The Relational Database .....	8
2.4 Database Management System (DBMS) .....	9
2.5 Web Database .....	10
2.6 Web Database integration .....	10
2.7 Web application tools .....	11
2.8 Study on similar systems .....	14
2.9 Technologies available .....	16

2.10 Summary .....	17
--------------------	----

## **CHAPTER 3**

<b>DESIGN OF SOLUTION</b> .....	18
3.1 Adopted methodology .....	18
3.2 Database Design and Development .....	18
3.3 Database Table contents .....	19
3.4 Input design .....	22
3.5 Prototyping .....	23
3.6 Unified Modeling Language (UML) .....	23
3.6.1 Use Case Diagram .....	24
3.6.2 Class Diagram .....	28
3.6.3 Activity Diagram .....	29
3.6.4 Sequence Diagrams .....	30
3.6.5 State Transition Diagram .....	31
3.7 Agile Software Development .....	32

## **CHAPTER 4**

<b>IMPLEMENTATION</b> .....	33
4.1 Selection of tools .....	33
4.2 Selection of Operating System (OS) .....	33
4.3 Selection of Programming Language .....	33
4.4 Summary of Minimum System Requirements .....	34
4.5 Selection of Database .....	34
4.6 Output Design .....	34
4.7 Description on main Modules .....	35
4.8 User Interface Design .....	36
4.9 Web Browser .....	37

## **CHAPTER 5**

<b>TESTING AND EVALUATION</b> .....	38
5.1 Introduction .....	38
5.2 Method of Evaluation .....	38
5.3 Evaluation Criteria .....	38
5.4 Evaluation results .....	39
5.4.1 Level of usage in Online Reservation for customers .....	43
5.4.2 Preference of Manual Vs Online reservation system by users .....	43
5.5 User-Friendliness of menus and interfaces .....	44
5.6 Usability of the proposed solution .....	44
5.7 Testing .....	45
5.8 Test Plan .....	45
5.9 Evaluation and analysis of feedback on project .....	50
5.10 Deviations from the project plan .....	52

## **CHAPTER 6**

<b>CONCLUSION AND FUTURE ENHANCEMENTS</b> .....	53
6.1 Future Enhancements .....	53
<b>REFERENCES</b> .....	55
<b>APPENDIX A – USER MANUAL</b> .....	56
<b>APPENDIX B – SOURCE CODES</b> .....	71

## List of Figures

Figure 2.1: User Interface (Home page) of Qloapps website .....	14
Figure 2.2: User Interface (Home page) of PHPTRAVELS website .....	15
Figure 2.3: User Interface (Home page) of Travelomatix website .....	16
Figure 3.1: High Level Use Case Diagram for Package Reservation System .....	25
Figure 3.2: Extended Use Case Diagram .....	26
Figure 3.3: Class Diagram .....	28
Figure 3.4: Activity Diagram for Package Reservation .....	29
Figure 3.5: Sequence Diagram for Package Reservation .....	30
Figure 3.6: State Transition Diagram for User ordering a package .....	31
Figure 4.1: Administrator functionalities .....	35
Figure 4.2: Login to Logout flow chart .....	36
Figure 4.3: Home page (User Interface) .....	37
Figure 5.1: Evaluation Questionnaire Invitation .....	40
Figure 5.2: Questionnaire .....	41
Figure 5.3: Usage Level of customers .....	43
Figure 5.4: Preference between Online vs Manual System .....	43
Figure 5.5: Evaluation feedback of User Interface .....	44
Figure 5.6: Login popup used for validation testing .....	49
Figure 5.7: Fields used for validations in customer registration form .....	50
Figure 5.8: Feedback on factors analysis chart .....	51
Figure A.1: Home page .....	56
Figure A.2: Login/Registration page .....	57
Figure A.3: View Packages page .....	58
Figure A.4: View User Profile page .....	59
Figure A.5: Add to Cart page .....	60
Figure A.6: View Cart page .....	61
Figure A.7: Payment/Checkout page .....	62
Figure A.8: Order Confirmation page .....	63
Figure A.9: Administrator Login page .....	64
Figure A.10: Dashboard tab (administrator module) .....	65



Figure A.11: Categories tab (administrator module) .....	66
Figure A.12: Package tab (administrator module) .....	67
Figure A.13: Customers tab (administrator module) .....	68
Figure A.14: Orders tab (administrator module) .....	69
Figure A.15: Settings tab (administrator module) .....	70

## List of Tables

Table 3.1: Users table .....	19
Table 3.2: User groups table .....	19
Table 3.3: User profile table .....	20
Table 3.4: Orders table .....	20
Table 3.5: Order options table .....	22
Table 3.6: Category products table .....	22
Table 3.7 : Use case narrative for Add Package use case .....	27
Table 5.1: Test Case format .....	46
Table 5.2: Example of Actual results format .....	46
Table 5.3 : High Level Acceptance Test Criteria .....	48
Table 5.4 : Test case execution table .....	49
Table 5.5: Figures of count from users'feedback .....	51

## List of Abbreviations

OTA	- Online Travel Agency
DBMS	- Data Base Management System
RDBMS	- Relational Data Base Mmanagement System
WWW	- World Wide Web
HTML	- Hyper Text Markup Language
HTTP	- Hyper Text Transfer Protocol
B2B	- Business to business
B2C	- Business to customer
CRM	- Customer Relationship Management
IBE	- Internet Booking Engine
SDLC	- Software Development Life Cycle
DBLC	- Data Base Life Cycle
UML	- Unified Modeling Language
OS	- Operating System
HCI	- Human Computer Interaction
GPS	- Global Positioning System

# CHAPTER 1

## INTRODUCTION

The Online Package Reservation System contains many features separated with different modules such as admin, customer. So the system manages the add/edit/delete (items or customers), customer management system, package bundling with selected items. The package bundling will be done according to the financial rules set by the organization (high cost, low cost etc) categorization.

Also, this will allow the customers and agents to create new reservations and to monitor them thereafter. This will help customers to manage their bookings easily and also to keep the data more safe. It will also help staffs to keep in track their customer's online booking request as well.

The management can easily access, retrieve the events that the customer reserves all the way he/she reserves something, manage the summary and report for the reservation of the event and it can add, edit and delete records.

The main purpose of this project is therefore to develop a web application program that would eliminate all problems encountered in the manual package booking system (mentioned below in problem definition section) and develop a web application with enhanced new features. So that customers can easily go online with their mobile phones, tablets or laptops in order to browse the relevant information they need about the Tours/holiday packages with their locality so that they can book for the appropriate suite that is within their budget.

### 1.1 Motivation

As a post-graduate student of Information Technology, I have been interested in this field since my career in computing began. With my experiences of working with Online Travel Agency (OTA) reservation system, course-work has led me to find new and latest features in online reservation systems.

This project has given me the opportunity to explore the world with parallel online reservation systems to find out new features and technologies to my system. My knowledge on technical side of this field at the beginning of the project was less, and I generally knew the high-level architecture on this system. As this is a subject that I'm working on my career, measuring the 'quality' of the system will be at a higher level. My previous basic knowledge of the Java language will cater for the development of this system up to some context, and my skills in testing measures, testing techniques, Issue tracking will be in higher level. When getting the suggestions from the community prior to develop this project, it was interesting and made a good impression on exploring new technologies and trends in the current market.

## 1.2 Problem Definition

In manual reservation system, they characterized with numerous problems. Some of these are customers having little or no information about the Tours within their journey, that is either too expensive or too unbefitting for his/her personality, a delay by the receptionist in retrieving certain information about any particular guest. All these problems and more would definitely make a Tour experience a down turn in business.

There are some drawbacks in doing a reservation through a manual reservation system such as:

- **Training problems:** the problems associated with training and advising people to be disciplined enough to maintain the maintenance system, i.e. to write the data into the system.
- **Report acquisition problems:** the effort associated with finding meaningful data and statistics in the system, registers and logs with good design are difficult, summarizing data and writing reports take lot of time.
- **Cost:** time will be money and employees.
- **Data duplication:** the same data gets repeated over and over since the workers find it hard to keep track of the documents, information and transactions.
- **Lack of security:** since data is stored in filing cabinets it is freely available to anyone. If information falls into the wrong hands it can be used against the company and customers and can blackmail them.
- **Common errors:** when entering data customers might have accidentally switched details and data since it is hand written.
- **Inconsistency of data:** there will be an unavailability for future use, since data might get misplaced during manual filing.so data won't be preserved properly for future use.
- **Repetition of work:** if there are any changes to be made, the data will have to be entered again.at times the worker would forget to make the changes or forget that they had already altered it and might redo it again, its again time consuming.
- **Too much paper work:** since everything and every details is written down manually in paper there will be too much paper work.
- **Space consuming:** since the data and paper is stored in filing cabinets it consumes too much place, as the amount of work done on paper increases the filing cabinets too increases.
- **Slow retrieval of data:** the information of customers and details are stored in different parts of the site and so takes a long time to retrieve the data.it takes a long time to find the information about a relevant person.in case of a Delivery, the delivery will be held back. This results in a sharp drop in sales, unhappy customers and a bad impression on the company.

## 1.3 Objectives

The main objective of this project is to reserve Packages based on Web technology. When a user make request from the web application, the system will carry out the following functionalities:

### Admin:

- Add product details
- Group the product for search option
- View/ Edit/ Register customers
- View payments for orders done by customers and change status of the reservation

### Customer:

- Search for product and information
- Reserve a package
- Allow customers to make payments and view the payment history details
- Customer registration
- Login/Logout functionalities

## 1.4 Project Significance

This online Reservation system will cater the society with the following advantages:

### 1. 24x7 open to make reservations

Many customers surf the Internet outside of business hours and they are more likely to make reservations on the spot that trying to remember to call back the next day. If normal business hours are 9 to 5 in the reservation company, then we have no one to take bookings after 17:00, and this is when we now know most online bookings occur. An Online booking system allows customer to receive booking 24 hours a day; therefore, the booking service will always be open.

### 2. Commission Free

Whether the reservation company is paying for direct advertising, or paying a commission for bookings through booking portals, in either case, we are paying a portion of the income to these sources.

If we have an online booking on our own website, we have then cut out the middleman.

### 3. Contain good customer service

Let's consider a contact form. A contact form does not provide the customers with a clear view of availability. A contact form does not allow the customers to book directly and pay and get confirmed automatically.

### 4. Minimize the workload

If we are not using an online booking system today, then we may still be processing all the bookings manually. A good online booking system will handle all the aspects of the booking. It will do the following:

- Automatically ensure that bookings can only be received when there's availability.
- Get all the information required during the booking process so agents don't have to waste time asking for more information
- Send out an automated email to the booking party as a confirmation of the booking

### 5. All customer data will be in a structured system

If we use a good online booking system which also handles the walk-in, email and phone bookings, then we will have all your previous customers' data in a structured system. This is important today as it allows to market the business to the previous customers. This customer list is often one of the best-valued resources in a business, and the business wants this data both accessible and structured.

## 1.5 Project Scope

The scope of this project is to deliver a system to reserve packages with combined items for a customer. This system caters packages from different locations based on low cost, medium cost, high cost categories etc. And the users can personalize the products they refer frequently through their web pages customized for them.

The solution of the proposed system will contain the features such as maintain and manage reservation system, reservation for packages, cancellation of bundled packages, pricing view and calculation through business rules, maintain the payment, multimedia support for the customer through image gallery to view the items ( park tickets, leisure events etc.) prior to complete the reservation, existing clients will be able to proceed with their usernames and if the client is a new, then he has to proceed by creating a new client profile, product items adding to cart level and validations, the payments were planned to be done through integrating a third party payment gateway or with the payments done by cash to the cashier at the property. (the cashier should have permission to access, edit, add and delete accounts details as an admin type)

The adding of new customers in the database will be done by filing up an application form

which will be recorded to the database and the proposed system will automatically generate an ID number that will also serve as proof that they are a registered user.

In file maintenance administrator can edit, update and add new offers to their system, updating of account user is also included.

After submitting the form a notification message will be displayed for the client confirming that the reservation is successfully made, the notification will include details about the reservation.

## 1.6 Flow of thesis

According to the content of this thesis, the **Chapter 1** described on the introduction, background, vision and the scope of the proposed project. In the process of developing the project, above mentioned factors were referred thoroughly and it could be align the project work with them during each phase of the process. This would be given a clear footprint on how and what kind of a system will be implemented.

In **Chapter 2**, it presents the initiatives taken to start the project. Gathering the correct functional and non-functional requirements related to the domain of the project was the base ingredient for all of the upcoming stages. Defining and organizing the architecture and technology to be adapt will also be included in this chapter. They will describe the following facts:

- How and what kind of a database to be selected.
- Which scripting languages to be developed.
- Detail description on the reference of similar systems

**Chapter 3** will discuss on detailed description of the solution design. It includes the methodology, database design and main tables included in it, design of input, prototyping, UML (Unified Modeling Language) diagrams to identify high level structure of the design etc.

In **Chapter 4**, I have mentioned on how the implemetation carried out in the process of the project. There, it contains detailed description on the tools used to develop, selection of OS, scripting language, database and the reason to select each of them. Then main modules of the system, overall picture of the system with logical flowcharts, and summarized details of User Interface design will be included.

**Chapter 5** consist with testing and evaluation process. Here, it describes the initial user categorization for requirement gathering, analysis of the data collected with graphs and percentages, testing, test plan, types of tests carried out with sample results and deviations from the initial project plan.



Finally **Chapter 6** will describe on the conclusion of the project with a summary and planned future enhancements where the system will be able to cater with a practical and commercial usage.

## CHAPTER 2

# REQUIREMENT ANALYSIS AND RELATED WORKS

### 2.1 Requirement Analysis

Many requirement gathering techniques were used to collect the actual requirements of the proposed system. Being an outside consultant of the current system was a significant advantage in the requirements gathering phase. The domain is more familiar and it was easy to meet users frequently and understand their problems.

The requirement gathering methods that were used are interviewing users, discussions, involving in the work, observation and studying the existing similar systems reports and documents.

By studying the existing similar projects, most important functions of a reservation system were identified. Also it was greatly helpful to clarify the user requirements. Some of those functions were good to be incorporated with the system. Also interfaces of those systems were very useful when designing the user friendly environment of the system.

The main users that directly involved in the package reservation system were identified as administrative staff, customers and visitors. Informal interviews and discussions were carried out with those key users. The expectations of the new system were mainly gathered from the community who willing to use online systems for their reservations.

Using all of those techniques the requirements were identified and analyzed to get the most important requirements of the proposed system to eliminate problems in the manual reservation system.

### Functional Requirements

- The system supports customers booking and able to modify them
- Customers can search based on different levels of package categories (ex. Location, high, medium, low cost packages)
- When a customer search for package and the search result must contain package information (Name, Image, Description and Price)
- Customers can book online and pay with credit or debit card.
- The system must notify booking confirmation message after successful payment.
- Customers should be able to create new accounts if they do not have an existing.
- Administrator should be able to add new packages under each category and delete existing.
- Administrator should be able to Group the product for search option
- Administrator is able to edit customers booking information (Order status etc.)

- Proper login and logout functionalities should be there according to the security features.
- Photo gallery should be up to date and subjective to the content
- There should be important contact details for a customer to contact in an emergency.

## Non-Functional Requirements

- System's scalability should be high to manage high amount of data (handle multiple transactions a time)
- The system must provide customers 24\*7 hours online booking service.
- The system should support almost all the browsers (Internet Explorer, Safari, Chrome, and Firefox).
- System should be user friendly and proper instructions should be given to the users in a convenient manner.
- System should be easily usable with simple interfaces because most of the users are non technical people.
- System should be more accurate and efficient and should not waste the time (response time should be quick)
- System should secure and protect sensitive data from unauthorized users.
- System's maintainability should be high and should be easy to be customized.

## 2.2 Reservation System

Nowadays in this generation every work is required to be done conveniently and efficiently use effort yet still having a productive job. With the use of online reservation information is more secure, more reliance and more accessible. Online Reservation system is used to store and retrieve information about product, product facilities and conduct transactions for booking it.

In today's growing market many businesses are turning toward reservation system to perform their everyday tasks. Online Reservation has important effects on business operations. No matter the size of your enterprise, in business it is a growing necessity. As the years go by, the business world is leaning more and more towards it. The role of technology in business caused growth in trade and commerce. The reservation in a specific good, Tour or package and paying bills can be done easily, no more hassles and consuming of time.

## 2.3 The Relational Database

The Relational Model developed out of the work done by Dr. F. F. Codd at IBM in the early 1970s. Its winning ways are particularly due to its conceptual simplicity and its ad-hoc query capability, At the core of the relational model is the concept of a table (also called a relation) in which all data is stored. Each table is made up of records (horizontal rows also known as tuples) and fields (vertical columns also known as attributes). Each table can be identified by behind the scenes. a unique name and that name can be used by the database to find the table behind the scenes.[1]

In the relational model, operations that manipulate data do on the basis of the data values themselves. This is done by comparing the value stored within a particular column for that row to some search criteria.

Another benefit of the relational system is that it provides extremely useful tools or database administration. Essentially, tables do not only store actual data but they also can be used as the means for generating meta-data, which is data about the table and field names which form the database structure, access rights to the database, integrity and data validation rules.

Thus, everything within the relational model can store in tables. This means that many relational systems can use operations recursively in order to provide information about the database. In other words, a user can query information concerning table names, access rights, or some data and the results of these queries would then be presented to the user in the form of a table. Relational database is the best choice for simple data storage because the technology has matured overtime and has continuously being improved to handle simple data effectively. The relational database environment is managed by a relational database management system, or RDRMS.

## **2.4 Database Management System (DBMS)**

Database management system is a software system that handles database management requirements such as create and maintain the database. DBMS is software that interacts with user's application programs and database. It allows adding, changing, deleting data, sorting and retrieving data from the database. Today, the most popular database management system is Relational database Management System (RDBMS). RDBMS concept was first described in 1970 by Dr. Edgar F. Codd in an IBM research publication called "System R4 Relational".[1]

A relational database uses the concept of linked two-dimensional tables which comprise of rows and columns. Database implementation is a way to put any database table on the web for viewing and/or editing with very little code. There are a number of RDBMS for both mainframe and microcomputer environments. Oracle, Sybase and Informix are multi user RDBMS.

MySQL is also one of the popular RDBMS available. On the other hand, Access, Foxpro, Paradox, Dbase and R:Base are RDBMS for microcomputer based. Categorically, RDBMS is the most commonly encountered in today's database and provides a great way to store and access complex information.

## 2.5 Web Database

The Web is the most popular application on the Internet. The Internet is rapidly changing the way information is generated, accessed and distributed. Furthermore, individuals and organizations across geographical and political boundaries find new and innovative ways to communicate, to cooperate in the sharing of information, and to collaborate in the creation of new works. Most new web applications response quickly, which is driving a dramatic shift from static content to data driven interactively dialog. Besides, the web application is able to transfer large amount of stored and organized data. Hence, it is clear that powerful and scalable databases are becoming an important component of web applications. The integration of databases and their components lead to the design of universal data access architecture, which enable companies to make operation more efficient and facilitate better and faster decision making. Thus, many database professionals believe that web database as a significant element in information system development.

## 2.6 Web Database integration

Web database application is an integration of database, web server, and web application server. A database as discussed above is a collection of data organized in a particular way. Databases can be of many types such as flat-file databases, Relational databases, object-oriented databases or multidimensional databases.

A client is a personal computer running a software application that can access data on a server and display them. The client side of the WWW is controlled by the Hypertext Markup Language (HTML) and the popular browser such as Netscape Navigator and Microsoft Internet Explorer.[1]

A server is a computer running software application that manages a data store containing: files of text, graphics formats, and multimedia data (video clips, and sound). Servers are set up by organizations to share information.

A Web server is an object that has internal state, whose persistence is typically achieved via a relational database management system. There are several popular web servers' applications available. An increasingly popular page server solution is the open source PHP. PHP is a server-side, cross platform scripting language similar to ASP that interfaces with the free MySQL database, as well as others. It is designed for developers with programming experience but has the advantage of being free and platform independent. PHP allows for rapid development of dynamic database-driven applications. [1]

The Web is good at displaying data and a relational database is good at managing data. Implementing a database driven Web site allows each tool to do what it does best. Making a Web site dynamic will completely transform the way information is delivered, what information can be delivered, and how it is delivered. The purpose of having a web database system to make the contents of an organization's database become available to the users. Essentially, all web applications do the same things.

## 2.7 The Web application Tools

Free Software refers to software distributed in source form which can be freely modified and redistributed. Apache, PHP and MySQL are free server type of software used in this project.

### Apache Web Server

The server side of WWW is controlled by Apache the workhorse of WWW network communication. Apache is an open-source and free web server software that powers around 46% of websites around the world. The official name is Apache HTTP Server, and it's maintained and developed by the Apache Software Foundation.

It allows website owners to serve content on the web, hence the name "web server". It's one of the oldest and most reliable web servers, with the first version released more than 20 years ago, in 1995. When someone wants to visit a website, they enter a domain name into the address bar of their browser. Then, the web server delivers the requested files by acting as a virtual delivery man.[2]

### Introduction on MySQL

MySQL is a fast, easy-to-use RDBMS used for databases on many Web sites. Speed was the developers' main focus from the beginning. In the interest of speed, they made the decision to offer fewer features than their major competitors (for instance, Oracle and Sybase). However, even though MySQL is less full featured than its commercial competitors, it has all the features needed by the large majority of database developers. It's easier to install and use than its commercial competitors, and the difference in price is strongly in MySQL's favor.

MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. [3]

### Advantages of MySQL

MySQL is a popular database with Web developers. Its speed and small size make it ideal for a Web site. Add to that the fact that it's open source, which means free, and you have the foundation of its popularity. Here is a rundown of some of its advantages:

- It's fast. The main goal of the folks who developed MySQL was speed. Consequently, the software was designed from the beginning with speed in mind.
- It's inexpensive. MySQL is free under the open source GPL license, and the fee for a commercial license is very reasonable.
- It's easy to use. You can build and interact with a MySQL database by using a few simple statements in the SQL language, which is the standard language for communicating with RDBMSs.

- Technical support is widely available. A large base of users provides free support via mailing lists. The MySQL developers also participate in the e-mail lists. You can also purchase technical support from MySQL AB for a very small fee.
- It's secure. MySQL's flexible system of authorization allows some or all database privileges (for example, the privilege to create a database or delete data) to specific users or groups of users. Passwords are encrypted.
- It supports large databases. MySQL handles databases up to 50 million rows or more. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- It's customizable. The open source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

## PHP Server Side Scripting Language

PHP is an "HTML-embedded scripting language" primarily used for dynamic Web applications. The first part of this definition means that PHP code can be interspersed with HTML, making it simple to generate dynamic pieces of Web pages on the fly. As a scripting language, PHP code requires the presence of the PHP processor. PHP code is normally run in plain-text scripts that will only run on PHP-enabled computers (conversely programming languages can create standalone binary executable files, a.k.a. programs). PHP takes most of its syntax from C, Java, and Perl. It is an open source technology and runs on most operating systems and with most Web servers. PHP was written in the C programming language by Rasmus Lerdorf in 1994 for use in monitoring his online resume and related personal information. For this reason, PHP originally stood for "Personal Home Page". Lerdorf combined PHP with his own Form Interpreter, releasing the combination publicly as PHP/FI (generally referred to as PHP 2.0) on June 8, 1995. [4]

### Characteristics in PHP

**Cross platform:** most PHP code can be processed without alternation on computers running many different operating systems. For Example, a PHP script that runs on Linux generally also runs well on windows.

**HTML-embedded:** PHP code can be written in files containing a mixture of PHP instruction and HTML code.

**Server-side:** The PHP programs are run on server-specially a web server.

**Web scripting language:** PHP programs run via a web browser.

## Introduction On JAVASCRIPT

JavaScript was originally developed by Netscape Corporation for use in its browser, Netscape Navigator. It includes a convenient syntax, flexible variable types, and easy access to the browser's features. It can run on the browser without being compiled; the source code can be placed directly into a Web page.

Modern JavaScript is a “safe” programming language. It does not provide low-level access to memory or CPU, because it was initially created for browsers which do not require it. In-browser JavaScript can do everything related to webpage manipulation, interaction with the user, and the webserver.

For instance, in-browser JavaScript is able to:

- Add new HTML to the page, change the existing content, modify styles.
- React to user actions, run on mouse clicks, pointer movements, key presses.
- Send requests over the network to remote servers, download and upload files (AJAX technologies).
- Get and set cookies, ask questions to the visitor, show messages.
- Remember the data on the client-side (“local storage”).

There are at least three unique things about JavaScript:

- Full integration with HTML/CSS.
- Simple things are done simply.
- Support by all major browsers and enabled by default. [5]

## The Advantages of JavaScript

**An Interpreted Language:** JavaScript is an interpreted language, which requires no compilation steps. This provides an easy development process. The syntax is completely interpreted by the browser just as it interpreted HTML tags.

**Embedded Within HTML:** JavaScript does not require any special or separate editor for programs to be written, edited, or compiled. It can be written in any text editor like Notepad, along with appropriate HTML tags, and saved as filename.html. HTML files with embedded JavaScript commands can then be read and interpreted by any browser that has JavaScript enabled.

**Design for Simple, Small Programs:** It is well suited to implement simple, small programs (for example, a unit conversion calculator between miles and kilometers or pounds and kilograms). Such programs can be easily written and executed at an acceptable speed using JavaScript. In addition, they can be easily interpreted into a web page.

**Performance:** JavaScript can be written such that the HTML files are fairly compact and quite small. This minimizes storage requirements on the web server and download time for



the client. Additionally, because JavaScript are usually include in the same file as the HTML code for a web page, they require fewer separate network accesses.

**Easy Debugging and Testing :** Being an interpreting language ,scripts in JavaScript are tested line by line, and the errors are also listed as they are encountered ,i.e. an appropriate error message along with the line number is listed for every error that is encountered. It is thus easy to locate errors, make changes, and test it again without the overhead and delay of compiling.

## 2.8 Study on similar systems

**Qloapps (Qloapps.com)** (<https://qloapps.com>) [6]

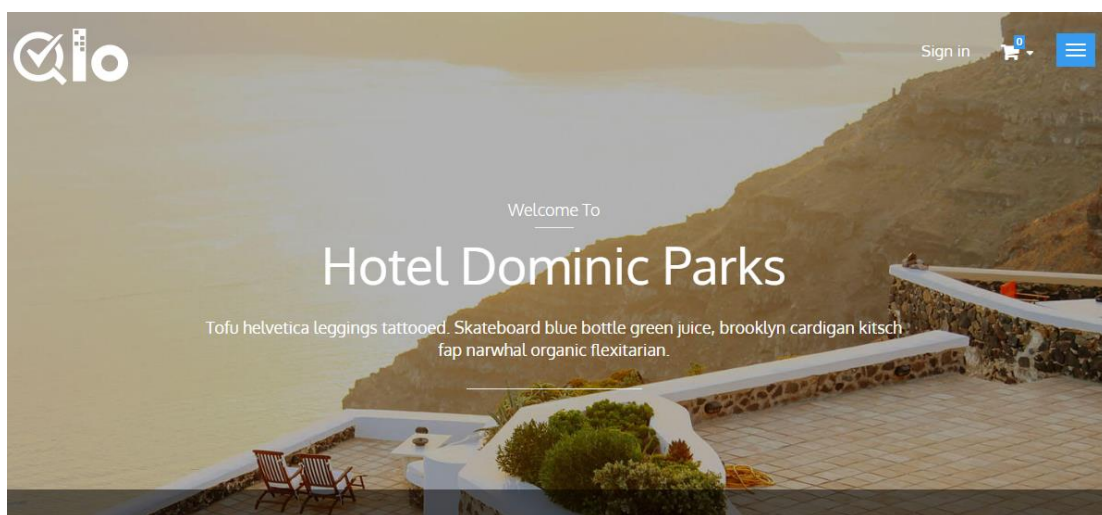


Figure 2.1: User Interface (Home page) of Qloapps website

This is one of the most famous online Tour booking systems available in the market and that too for free of cost.

It is an open-source and independent software which helps Touriers easily launch their own Tour booking website. You can attract the attention of online guests through by displaying compelling pictures of the Tour, interior and amenities on the beautiful landing page. It's quick and simple booking process delivers a unique guest experience to the customers. It also provides a single platform to easily manage on-desk and online bookings.

Qloapps has many features like Booking Order Restriction, Order Edit, Partial Payment Option, Partial Booking, Responsive theme, Google Maps, Support for Multiple Languages and Currencies and many more.

## PHPTRAVELS-Travel Technology Partner

(<https://www.phptravels.net>) [7]

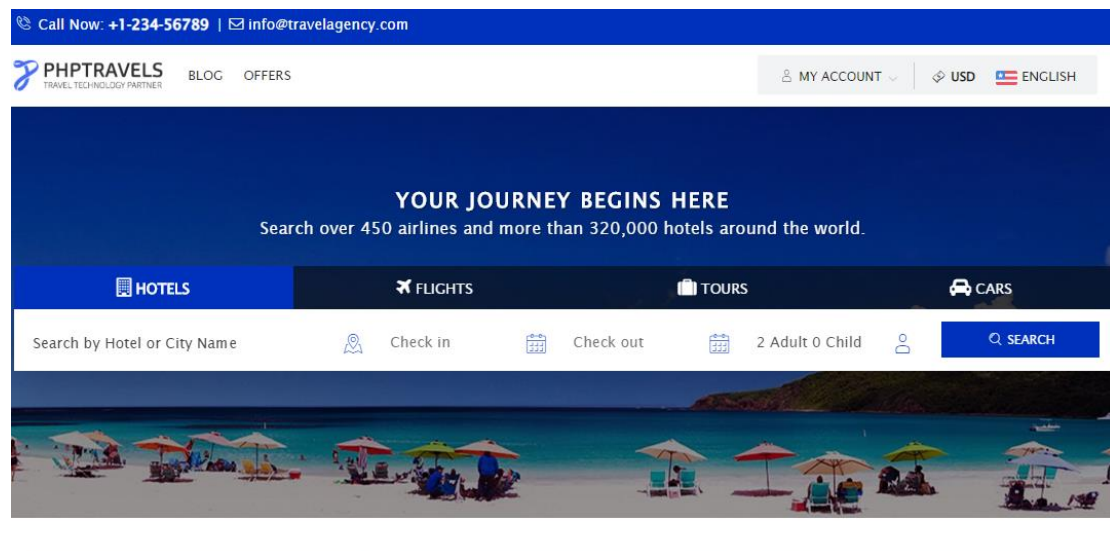


Figure 2.2: User Interface (Home page) of PHPTRAVELS website

PHPTRAVELS providing the best and user friendly application to our customers, with extensive tools specially for online travel business from hotels booking to flights reservation, we also provide custom web solutions and services. With over 4 years of experience phptravels.com have unquestionably won a reputation for being a trusted source, a reliable partner and an expert in the area of online travel business applications.

The combination of our services, 24/5 nonstop support, our pricing, friendly way of conducting business, and their compassionate corporate philosophy is very unique in today's business world. They provide exceptional customer service and strive to build lasting relationships with their customers by making it easy and profitable for them to do business with them. They value their clients and focus in customer satisfaction. That is why after-sale support is a key factor in their approach. With all this rich experience, now through PHPtravels they are bound to meet customer satisfaction and establish trust.

## 2.9 Technologies available

### Travelomatix

(<http://www.travelomatix.com>) [8]



Figure 2.3: User Interface (Home page) of Travelomatix website

Top travel technology solutions should consist the travel software and mobility solutions packaged into a single offering. It's important that travel software support the mobile browsers as mobile devices are bringing now more that 30% of the online bookings. Leading travel software are offering flights, Tours, transfers, sightseeing and dynamic packaging.

Travelomatix is an award winning travel technology company, delivering B2B / B2C travel software, travel CRM, accounting software, car rental software and mobility solutions to global travel & hospitality companies. They work with over forty top travel aggregators, global distribution systems and channel managers. With clear focus on technology, agile methodology and innovation, it ensure quality deliverables and speedy turnaround to develop and deliver scalable travel portals and booking engines.

### Adapted technologies in high level

- Revolutionary booking engine platform lets you engage B2C users and B2B sub agencies to make more sales.
- Use your own domain name, or purchase new one through Travelomatix. Sell the flights, hotels and other travels deals under your brand.
- Cloud-based IBE (internet booking engine) is 1/8 the cost of a custom travel booking engine with equal features.
- Travelomatix is 100% mobile ready. Travel website will look and perform equally well on desktops, tablets and mobile phones with no extra coding or design work.

- Travelomatix offers payment gateway integration of your choice. Sell online deals on your travel website and receive payments directly into your account.
- Integrate travel booking engine with leading enterprise resource planning suites like NetSuite, Brightpearl and Microsoft Dynamics.

Travelomatix engine is developed with high end technology to deliver the highest quality operations.

## **2.10 Summary**

As web applications become increasingly integrated in online reservation, the need to build reliable, usable and adaptable systems has become significant. The integration of databases and their components enable companies to make operation more efficient and facilities better and faster decision-making. Thus, web database has become a significant element in today's reservation system development.

In this study, a web based package reservation system project will be developed based on the planning, analysis and a full understanding on the areas that needed to be emphasized in the development of database driven reservation system using Apache, PHP and MySQL free software.

## CHAPTER 3

### DESIGN OF SOLUTION

This chapter discusses the approach used in defining a web based Reservation System. The discussion covers the System Development Life Cycle (SDLC) and Database Life Cycle (DBLC), Agile methodologies and Unified Modeling Language (UML) for diagramming technique.

#### 3.1 Adopted methodology

SDLC is a logical process used to develop the reservation system, including requirements, validation, training, and user interface design. Any SDLC should result in a high quality system that meets or exceeds customer expectations, reaches completion within time and cost estimates, works effectively and efficiently in the current and planned Information Technology infrastructure, and is inexpensive to maintain and cost effective to enhance. It adheres to important phases that are essential for development, such as planning, analysis, design, and implementation.

For many phases, I dopted Agile process in requirement gathering, parallel run of testing, development and implementation with the changing specifications.

#### 3.2 Database Design and Development

Database system is a fundamental component of any reservation system. The standard Database Life Cycle (DBLC) procedure will be used in the development of the package reservation system. The stages of the database application life cycle are shown below.

The various steps in this procedure are conceptual, logical and physical design.

In the **conceptual design**, the objectives and requirements of the database are defined. Then the entities, attributes and the relationships that exist between entities are identified. The attributes will be assigned to entities through the process of normalization in order to reduce data redundancies and to eliminate data anomalies.

The **logical design** then translates the conceptual design into an internal model for the database. It includes the design of tables, indexes, views, transactions and others.

At the **physical design** stage, the data storage and data access characteristics of the database would be selected in order to ensure maximum performance of the database. The development of the package reservation system will involves the database initial study, database design and prototyping.

### 3.3 Database Table contents

Following are some of the tables which contains in the database design.

**1) Table name : users**

**Description :** It is used to store the user login details like username, password and group id (used as access level).

Field	Type	Null	Default	Comments
user_id	int(11)	Not null		Primary key
group_id	int(11)	Not null		Group id
user_email	varchar(100)	Not null		Users mail id
user_password	varchar(40)	Not null		User password
user_status	tinyint(1)	Not null	Null	User status
user_approved	tinyint(1)	Not null	Null	User approved
user_created	int(11)	Not null		User created
last_login	int(11)	Not null		Last login
last_ip	varchar(40)	Not null		Last IP

Table 3.1: Users table

**2) Table Name : User groups**

**Description :** It is used to store user types

Field	Type	Null	Default	Comments
group_id	int(11)	No	Null	Group id
group_name	varchar(20)	No	Null	Group Name
group_description	varchar(100)	No	Null	Group Description

Table 3.2: User groups table

**3) Table Name** : user profile

**Description** : It is used to store the all user profile.

Field	Type	Null	Default	Comments
profile_id	int(11)	No		Profile id
user_id	int(11)	No		User id
first_name	varchar(50)	Yes	Null	First Name
last_name	varchar(50)	Yes	Null	Last Name

Table 3.3: User profile table

**3. Table Name** : Orders

**Description** : It is used to store the all orders.

Field	Type	Null	Default	Comments
order_id	int(11)	No		Order id
user_id	int(11)	No		User id
first_name	varchar(32)	No		First Name
last_name	varchar(32)	No		Last Name
customer_email	varchar(100)	No		Email
customer_telephone	varchar(32)	No		Telephone
shipping_company	varchar(32)	No		Company
shipping_first_name	varchar(32)	No		Shipping
shipping_last_name	varchar(32)	No		Shipping
shipping_address	varchar(128)	No		Address
shipping_post_code	varchar(10)	No		Post code
shipping_city	varchar(128)	No		City
shipping_country_id	int(11)	No		Country

Field	Type	Null	Default	Comments
shipping_zone	varchar(128)	No		Zone
shipping_method	varchar(128)	No		Company
payment_company	varchar(32)	No		Payment
payment_first_name	varchar(32)	No		Payment
payment_last_name	varchar(32)	No		Name
payment_address	varchar(128)	No		Address
payment_post_code	varchar(10)	No		Post code
payment_city	varchar(128)	No		City
payment_country_id	int(11)	No		Country id
payment_zone	varchar(128)	No		Zone
payment_method	varchar(128)	No		Method
comment	Text	No		Comment
total	decimal(17,2)	No	0.00	Total
shipping_cost	decimal(17,2)	No	0.00	Cost
tax_description	varchar(255)	No		Description
tax_rate	decimal(17,2)	No	0.00	Tax rate
tax_shipping	int(1)	No	0	Shipping
currency	varchar(20)	No		Currency

Table 3.4: Orders table

1) **Table Name** : order\_options

**Description** : It is used to store the all order options details.

Field	Type	Null	Default	Comments
order_option_id	int(11)	No		Option id



Field	Type	Null	Default	Comments
order_id	int(11)	No		Order id
order_product_id	int(11)	No		Product id
option_value_id	int(11)	No		Value id
option_name	varchar(255)	No		Name
option_value	varchar(255)	No		Value
option_price	decimal(17,2)	No	0.00	Price
option_type	char(1)	No		Type

Table 3.5: Order options table

2) **Table Name** : category\_products

**Description** : It is used to store the all products details.

Field	Type	Null	Default	Comments
product_id	int(11)	No		Product id
category_id	int(11)	No		Category id

Table 3.6: Category products table

### 3.4 Input design

The user interface design is very important for any application. The interface design describes how the software communicated within itself, to system that interpreted with it and with humans who use it. The interface is a packing for computer software if the interface is easy to learn, simple to use. If the interface design is very good, the user will fall into an interactive software application.

The input design is the process of converting the user-oriented inputs into the computer-based format. Errors entered by data entry operations can be controlled by input design. The data is fed into the system using simple interactive forms. The forms have been supplied with messages so that user can enter data without facing any difficulty. The data is validated wherever it requires in the project. This ensures that only the correct data have been incorporated into the system. The goal for designing input data is to make data entry as easy, logical and free from errors.

The objectives of input design are:

- To produce a cost effective method of input
- To make the input forms understandable to the user
- To ensure the validation of data input
- To achieve the highest position level of accuracy

The various activities to be performed for the overall input processors are:

- Data recording at its source.
- Data transfer to input form.
- Data conversation to computer acceptable mode.
- Data validation.
- Data flow control.
- Data correction if necessary.

### **3.5 Prototyping**

The prototype was developed using the relational database in the web environment. In this project, Apache open source software has been selected. For the database, MySQL a popular relational database server application for building web applications is chosen. PHP a server side, cross-platform scripting language will be used as a middleware. To create and maintain MySQL databases, PhPMyAdmin a web-based interface to MySQL is used. phpMyAdmin handles the administration of MySQL over the web. It can create and drop databases, create, copy, drop and alter tables, delete, edit and add fields. It also can execute any SQL-statement, manage keys on fields, load text files into tables, create and read dumps of tables.

### **3.6 Unified Modeling Language (UML)**

Today systems analysis and design are moving toward object-oriented techniques. Using these techniques, one can view a system as a collection of self-contained objects that have data and process. The object approach is implemented using Unified Modeling Language (UML) notation. UML is a diagramming technique to model system development project. The consistent notation, integration among diagramming techniques, and the application of the diagrams across all phases make UML a powerful language in many organizations system analysis and design methodologies. UML defines a set of several object-diagramming techniques used to model a system as described below.

#### **Use case diagram**

The interaction between users and the system

#### **Class diagram**

The static nature of a system at the class level

#### **Sequence diagram**

The interaction between classes for a given use case, arranged by time sequence.

## **State chart diagram**

Sequence of states that an object can assume, the events that cause an object to transition from state to state and significant activities an action that occur as a result.

## **Activity diagram**

A specific business process or the dynamics of a group of objects, provides a view of flows and what is going on inside a use case or among several classes.

The four main UML diagramming techniques that dominate object-oriented projects are Use case diagrams, Sequence diagrams, Class diagrams and State transition diagrams.

### **3.6.1 Use Case Diagram**

Use cases are the primary drivers for all the UML diagramming techniques. Use case diagrams give an outsider's view of a system. The use case communicates that what the system must do at a high level. They describe the behavior of a system from user's point of view by using actions and reactions by the actors. They define the system's boundary and the relationships between the system and the environment.

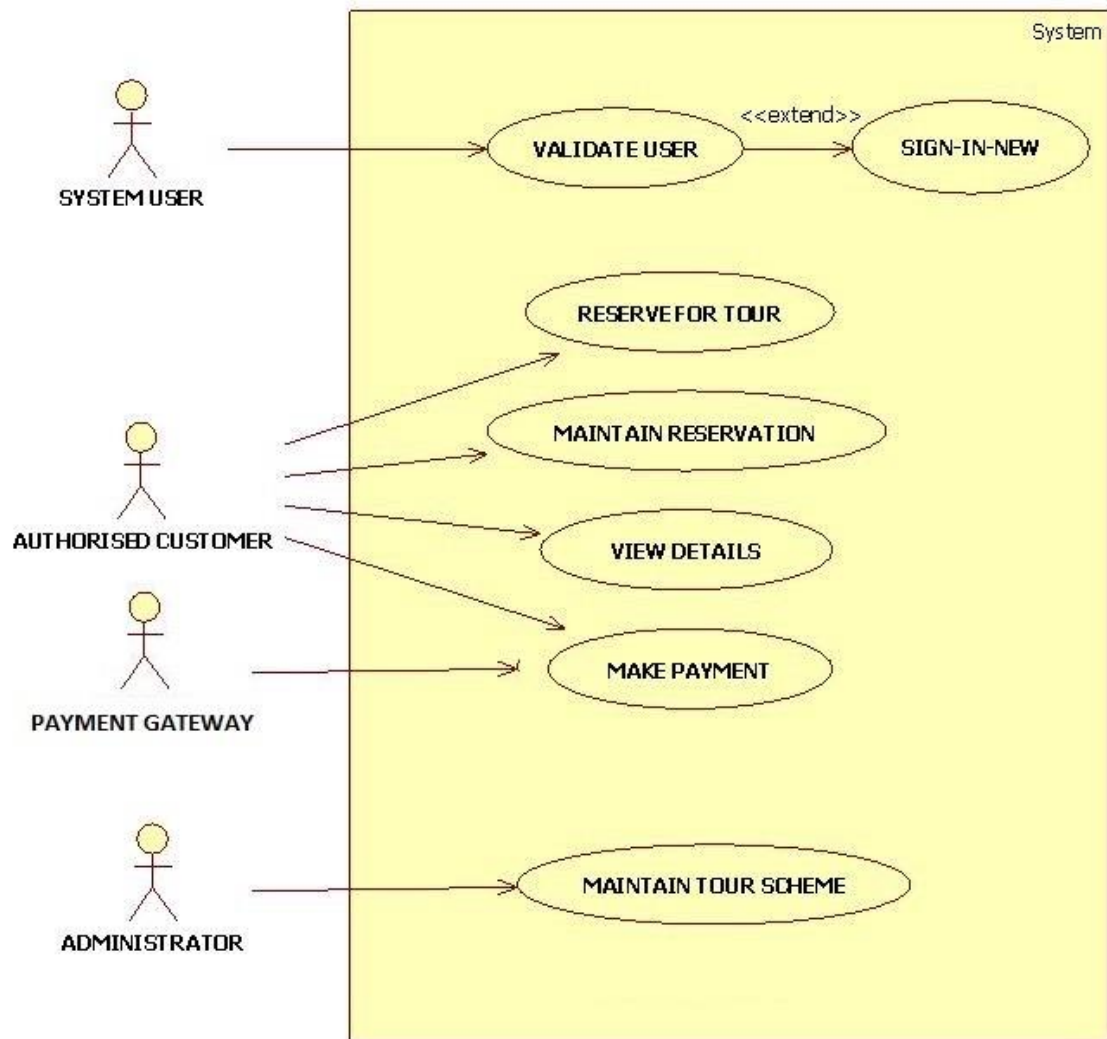


Figure 3.1: High Level Use Case Diagram for Package Reservation System

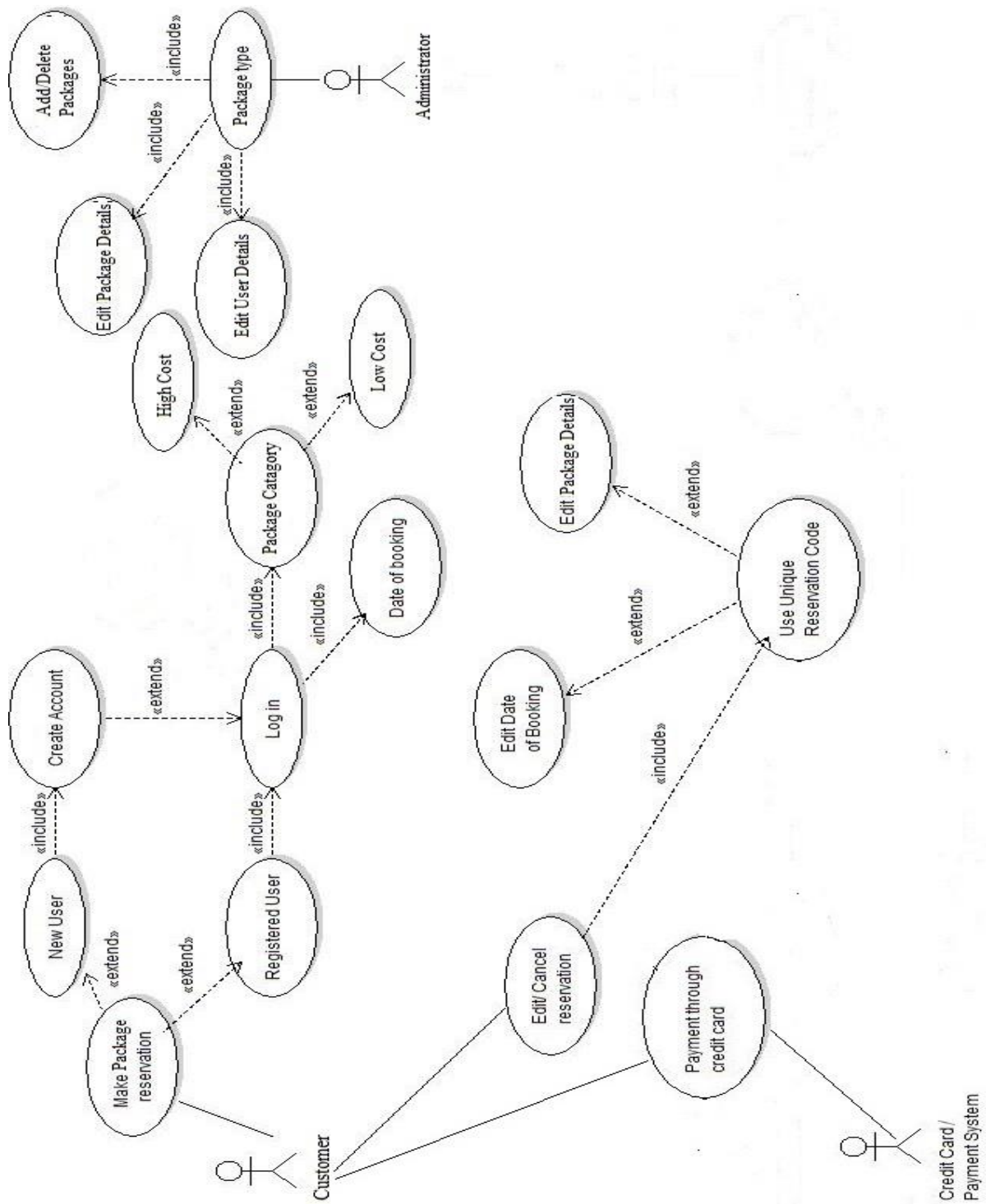


Figure 3.2: Extended Use Case Diagram

## Use Case Description

Following table shows the use case narrative for the Add package by the actor Customer.

Use Case Name:	Adding high cost package to cart and confirm the order
Use Case ID:	UC001
Priority:	High
Source:	Valid customer registration form
Primary Business Actor:	Customer
Trigger:	Select action
Preconditions:	Registered customer needs to login to the system with the correct user name and the password
Main Scenario:	<ol style="list-style-type: none"> <li>1. Navigate to the home page</li> <li>2. Select package type (High cost package)</li> <li>3. Package description will appear.</li> <li>4. Select the add to cart option and view the cart.</li> <li>5. Add/Edit/delete cart</li> <li>6. Navigate to payment page, select method.</li> <li>7. Confirm reservation</li> </ol>
Post Condition:	New order will be added for the registered customer.
Exception Path:	If there is a connection failure, the web server returns to the wait state.

Table 3.7 : Use case narrative for Add Package use case

### 3.6.2 Class Diagram



Figure 3.3: Class Diagram

### 3.6.3 Activity Diagram

This includes dividing and separation of the diagram in to logical streams using 'swim lanes'.

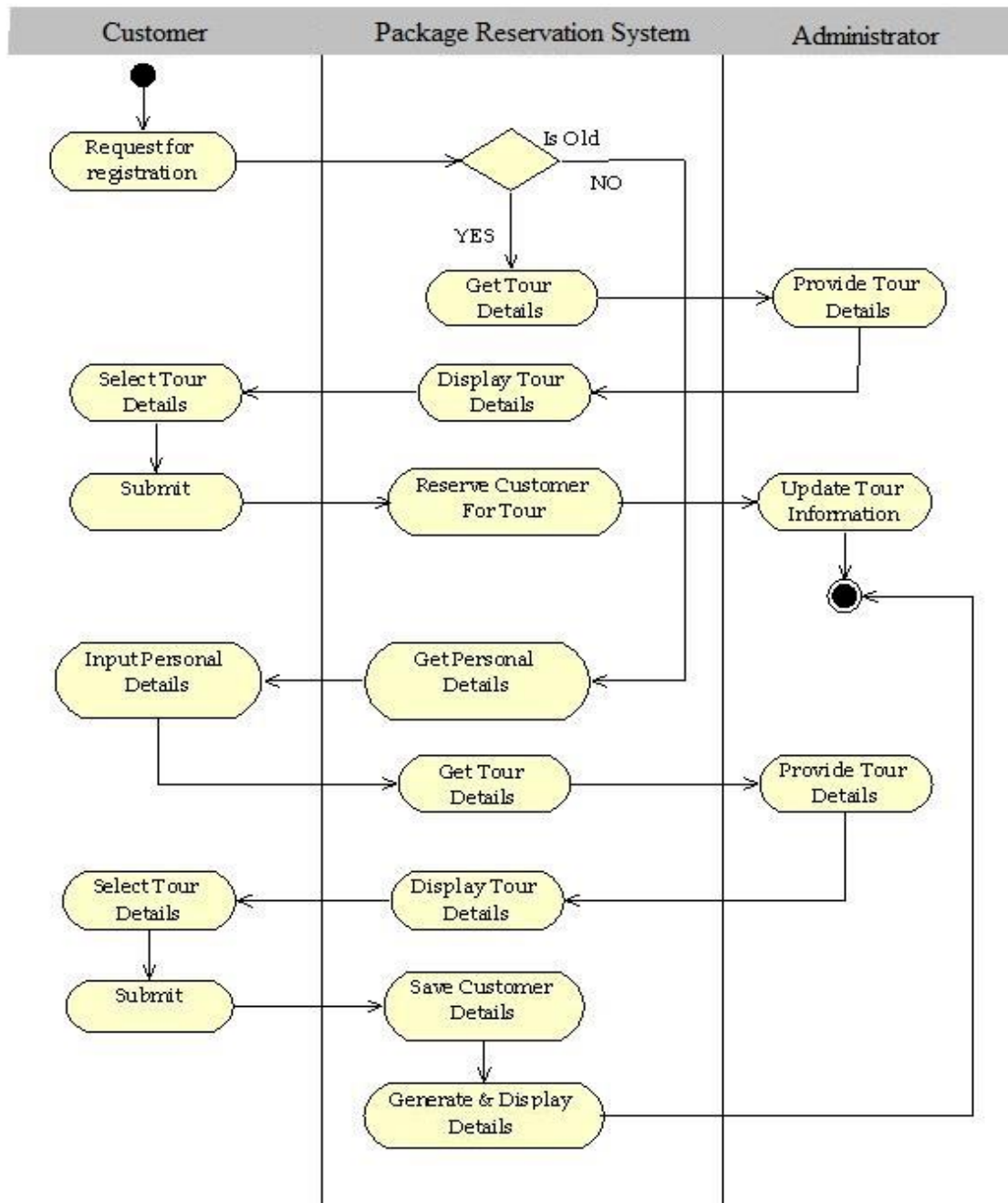


Figure 3.4: Activity Diagram for Package Reservation



### 3.6.4 Sequence Diagrams

This UML diagramming technique illustrates the classes that participate in a use case and the message that pass between them over the time.

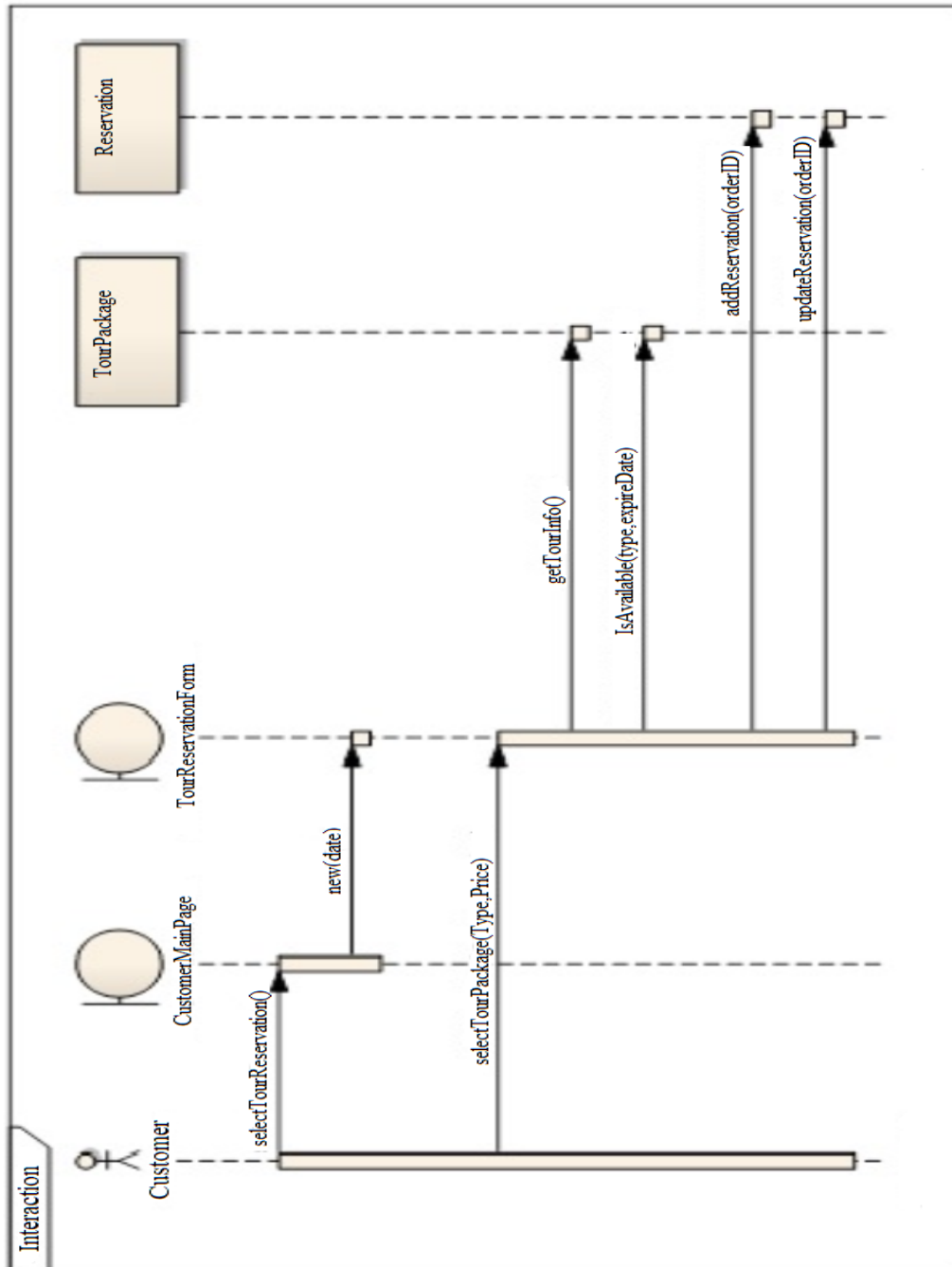


Figure 3.5: Sequence Diagram for Package Reservation

### 3.6.5 State Transition Diagram

This diagram shows the states of a user. According to the diagram a user can be a Customer or an Administrator/ Agent(on behalf of a customer). And Guests(Visitors) are also there in the system.

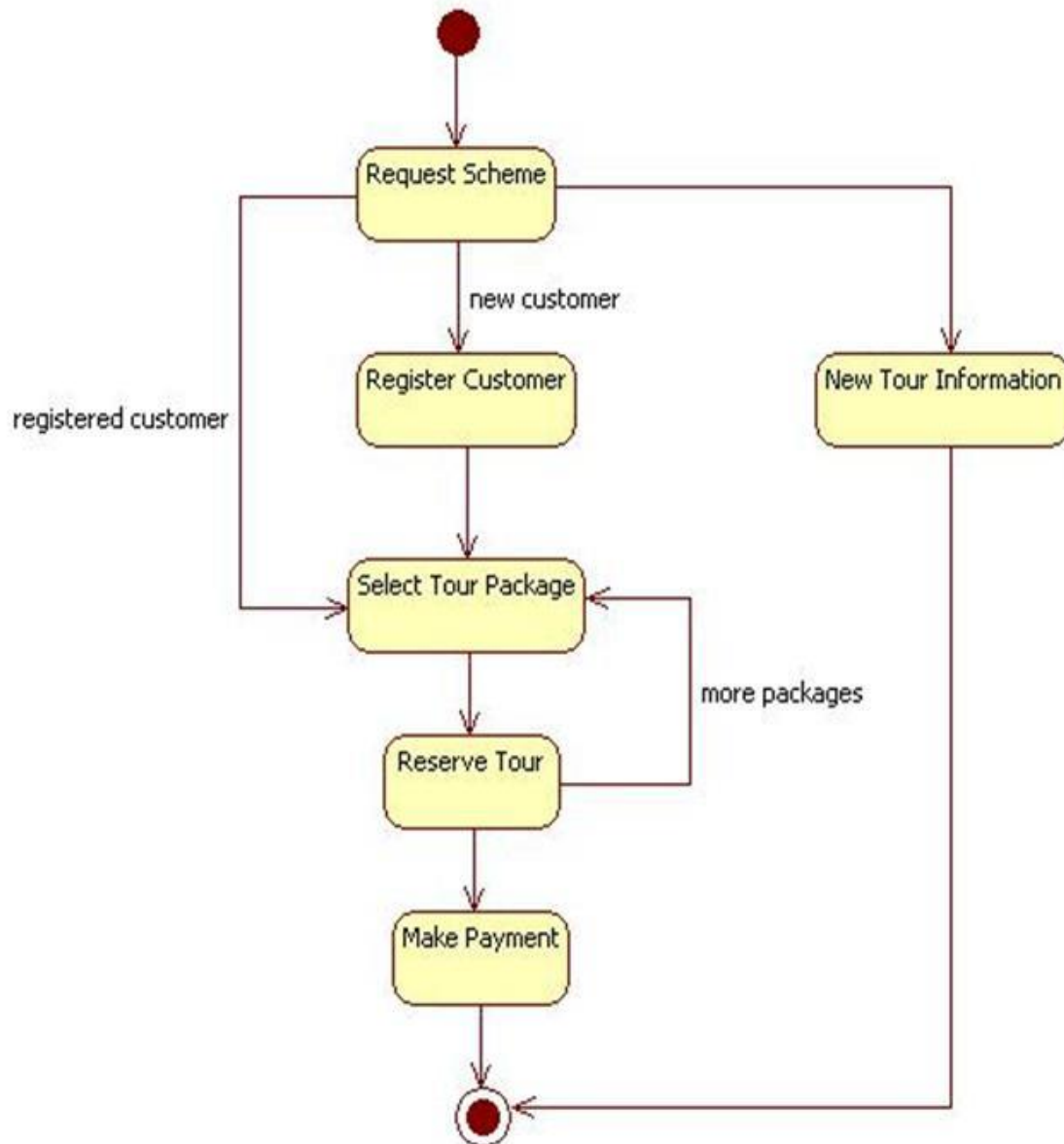


Figure 3.6: State Transition Diagram for User ordering a package

## **3.7 Agile Software Development**

For this project Agile methodologies were best suited to implement. There were significant changes to the initial requirements and the mock-up designs and the functionalities were changed a lot. With this process, most of the times the analysis, design and implementation phases were happened concurrently.

Initially, a prototype was developed using the relational database in the web environment. MySQL, a popular relational database server application was used as the database for building web applications. PHP, a server-side cross-platform scripting language was used as a middleware.

# CHAPTER 4

## IMPLEMENTATION

### 4.1 Tools Selection

It is much important to select proper implementation tools to accept the system by the clients. Generally, this will be the first step in any software implementation process. When selecting the development platform it includes the selection of hardware and software, will will be mandatory to build the reservation system.

System developer has to consider many aspects to enhance the functionality and performance of the system. Therefore this section explains the necessary concerns regarding the selection of tools which are used to develop and run the system. System developer has to select the operating system initially. Then the selection of the development programming language and Database Management System was done. As this is a web based system, it was needed to think about the server side as well. Then need to maintain a wb server with a suitable operating system. Finally, it was decided on the required software and hardware configurations for current client requirements to implement the system properly.

### 4.2 Selection of Operating System (OS)

To run the client side programs Microsoft Windows XP was used as to maintain as minimum requirements. At the development stage it was not needed to consider on the server side as this was run and tested on localhost. But it should be properly implemented in server side as well.

When selecting the OS, the following factors were mainly considered:

- Security
- Reliability and Stability
- Performance and Resource management
- Scalability
- Maintainability
- 

### 4.3 Selection of Programming Language

In the development phase, PHP and Javascript was mainly used as programming languages while considering the following factors:

- Familiarity
- Quick and easy communication with Databse
- Execution Speed and Efficiency
- Error handeling
- Standardization

- 

## 4.4 Summary of Minimum System Requirements

### Hardware Requirements:

- Processor Name : Dual Core
- Processor Speed : 3.2 GHz
- RAM : 1 GB
- Hard Disk Capacity : 40 GB
- Display Device : 14' to 19' Inch Monitor
- Keyboard Type : PS2 or USB
- Mouse Type : PS2 or USB

### Software Requirements:

- Technology Implemented : Apache Server , Dream weaver
- Language Used : PHP 5.2
- Database : My SQL 5.2
- User Interface Design : HTML, AJAX , JAVA SCRIPT
- Web Browser : Chrome,Mozilla,IE8
- 

## 4.5 Selection of Database

Storing and retrieving data are critical aspects in the system development and measurements in performance. Here, MySQL was selected as the Database Management System. Following factors were considered when selecting the above option:

- Well defined DBMS
- Ease of use
- Following standards
- 

## 4.6 Output Design

Computer outputs are the most important and direct source of information to the user. A quality output is one which meets the requirements of the end user and which presents information in a way which is clear, easy to read and visually attractive. The screens are designed in such a way that the outputs are provided to the user in an understandable form.

The objectives of output design are:

- Design output to serve the intended purpose.
- Provide output on time.
- Assume that output is where it is needed.
- Design output to fit the user

## 4.7 Description on main Modules

### Administrator Module

- **Profile view** – Admin can view out top five categories and top five products.
- **Categories** – Admin can create and add new categories.
- **Products** – Admin can view all product and description.
- **Customer** – Admin can view, add and edit customer's details.

### Flow Chart [ADMIN]

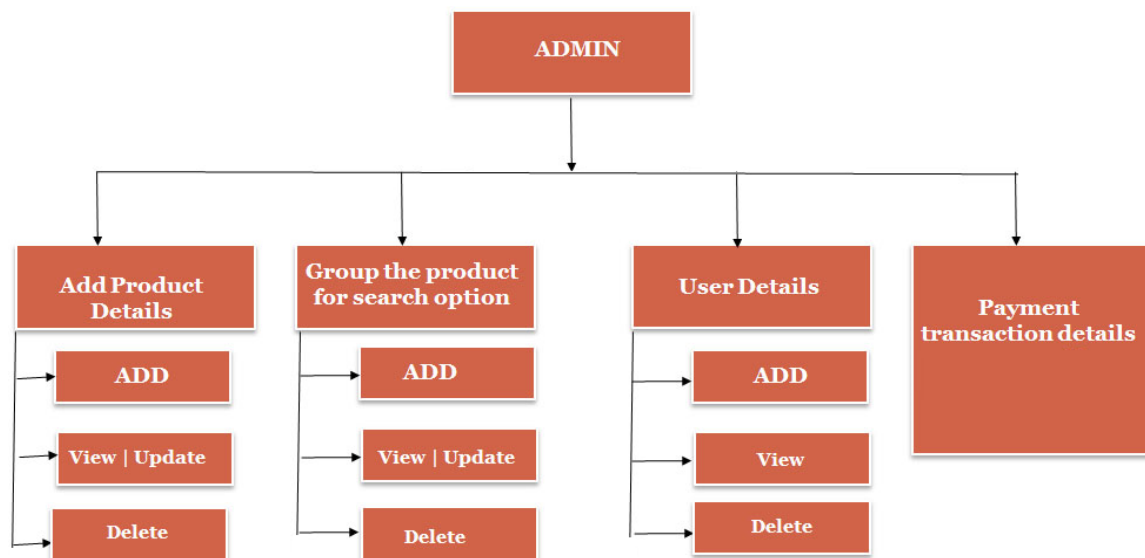


Figure 4.1: Administrator functionalities

### Customer Module :

- **View** – User can view cameras, computer and laptop.
- **Buy** – User can buy any item first go to read more and add to cart.
- **Summary** – User can view shopping card summary and used coupons.
- **Check out** – User click check out show total details

## Security Policy

The Security policy assigned by admin consist of access level for user login and functionality chance for registered users and visitors.

## Login & Logout

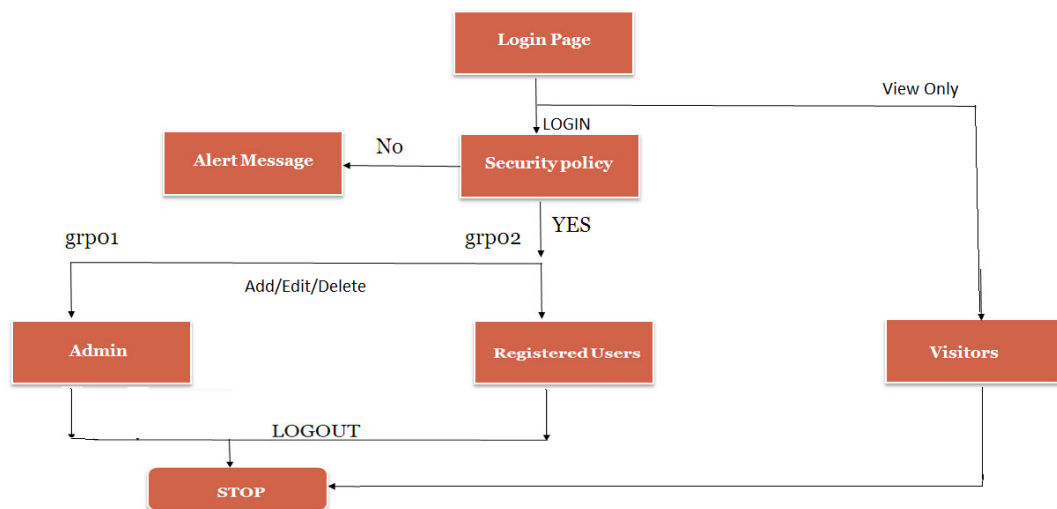


Figure 4.2: Login to Logout flow chart

## 4.8 User Interface Design

Today with the web technologies, User Experience is more prominent than the User Interface. This design will communicate the business domain to the users and this will be a major component in Human Computer Interaction (HCI). User's first impression will rely on this factor with the characteristics in usability. Here forms in PHP, has been mainly used to retrieve the user's responses. User Interface (UI) design has been made through HTML, AJAX, JAVA SCRIPT technologies.

Below are some of the User Interfaces which will be described further in next chapters:

## USER HOME PAGE

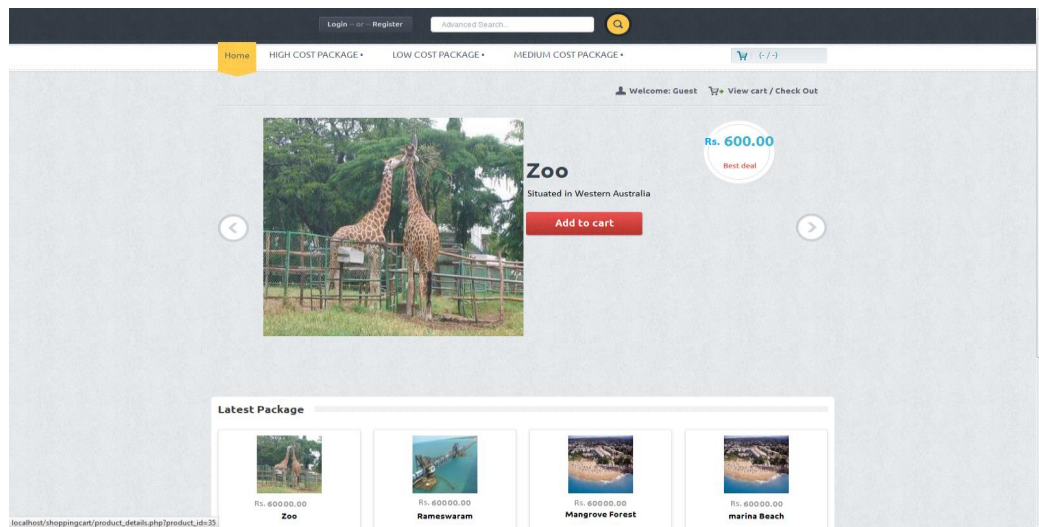


Figure 4.3: Home page (User Interface)

## 4.9 Web Browser

As this project runs on web based, it is important to consider the Web Browsers. Here mainly Internet Explorer has been used in development phase and it is commonly tested through Google Chrome.



## **CHAPTER 5**

# **TESTING AND EVALUATION**

### **5.1 Introduction**

This section will discuss on the evaluation of the project, whether the initial goals of the project were achieved. At this final stage it is essential that the prototype is critically evaluated in order to decide whether the project is a success or a failure.

First section is the critical evaluation of the entire project based on achieved project objectives. The second section consists of an evaluation of the prototype by users. The third section includes a critical assessment of the project.

The current chapter explains the evaluation process chosen by the author to critically evaluate the prototype.

### **5.2 Method of Evaluation**

The evaluation of the prototype was carried out by checking whether prototype has achieved objectives mentioned on the project proposal and it's practical usage. Following are the selected types of users for the evaluation:

- Expert user
- Ordinary users

The selected method of evaluation was to present the prototype to above mentioned users through a demonstration. Then the feedback was analyzed through the responses received to the questionnaire issued to the users.

### **5.3 Evaluation Criteria**

Evaluation was carried out to obtain opinions, remarks and suggestions about the product. Several techniques were user for evaluation. These included interviewing, questionnaires and demonstrations.

#### **Review of objectives**

The main objective of this project is to reserve Packages based on Web technology. When a user make request from the web application, the system will carry out the following functionalities:

Admin:

- Add product details
- Group the product for search option
- User details
- Payment transaction details

Users:

- Search for product and information
- Reserve a package
- Payment transaction details
- User registration
- Login/Logout functionalities

**Maintenance:**

- Maintenance of user log:

The system provide user maintain facilities to the administrator. Registration request can be approved by filling all required details in the registration form. In addition to that user deletion and editing can be done.

- Maintenance of Website:

Maintaining packages (updating, delete existing or adding new packages) has been provided to the administrator in order to carryout efficient website service.

**5.4 Evaluation results**

Evaluation results are based on the questionnaires distributed among the community. The following notice was distributed among people to give an awareness of the system.

## Tour Package Reservation System

### Evaluation Questionnaire

The aim of this questionnaire is to build an **Online Tour Package Reservation System** for Holiday Packages.

The applicability and usefulness of the proposed system will be assessed through the information that will be generated through this questionnaire.

Therefore I welcome your feedback.

**Thank You!**

R.P.U.L. [Rajapaksha](#)

Masters in IT

(University of Colombo School of Computing)

Email: [udula.rajapaksha@gmail.com](mailto:udula.rajapaksha@gmail.com)

Figure 5.1: Evaluation Questionnaire Invitation

Following is the details of the questionnaire distributed:

1. Name <input type="checkbox"/> Mr. <input type="checkbox"/> Mrs. / Miss.....
2. Age group <input type="checkbox"/> 16 - 20 <input type="checkbox"/> 21 - 30 <input type="checkbox"/> 31 - 50 <input type="checkbox"/> Above 50
3. Occupation/ Profession .....
4. How would you rate your knowledge of using web literacy? <input type="checkbox"/> An advanced user <input type="checkbox"/> An average user <input type="checkbox"/> Basic user <input type="checkbox"/> Non-literate
5. Have you used any online package reservation software before? <input type="checkbox"/> Yes <input type="checkbox"/> No  If Yes, please specify the purpose? ..... .....
6. Prefer online package reservation compared to manual method? <input type="checkbox"/> Yes <input type="checkbox"/> No
7. Was the graphical user interface easy to use? <input type="checkbox"/> Yes <input type="checkbox"/> No

8. Level of satisfaction?

Very satisfied

Satisfied

Unsatisfied

Very unsatisfied

Neither

Any comments

.....

.....

9. Suitability of online package reservation system for other organizations?

Yes

Can be

Not Sure

No

Any comments

.....

.....

10. Further improvements / enhancements that you feel necessary to make this software more useful?

.....

.....

Figure 5.2: Questionnaire

The majority of the users who evaluated this project were of the opinion that web application package system has a great business importance for Tour Packages to carry out their operations in an efficient manner. By using this system, they can improve the Tour Package services and competitiveness among the business domain. The following evaluation results were based on user's comments obtained during the evaluation process.

### 5.4.1 Level of usage in Online Reservation for customers

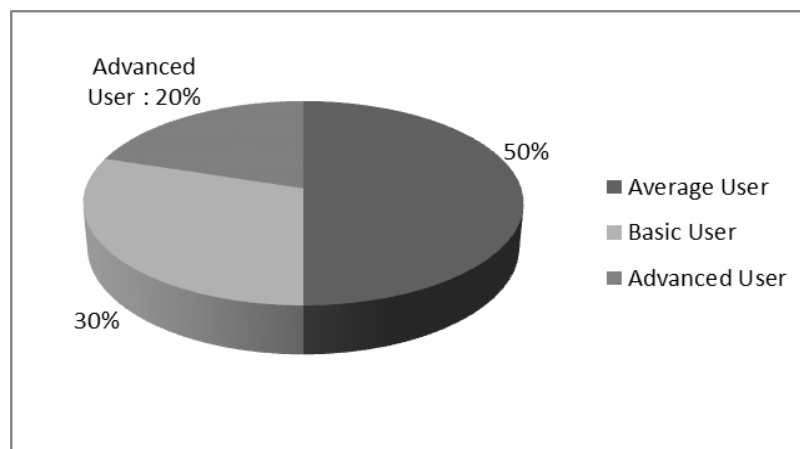


Figure 5.3: Usage Level of customers

### 5.4.2 Preference of existing manual reservation system Vs Online reservation system by users

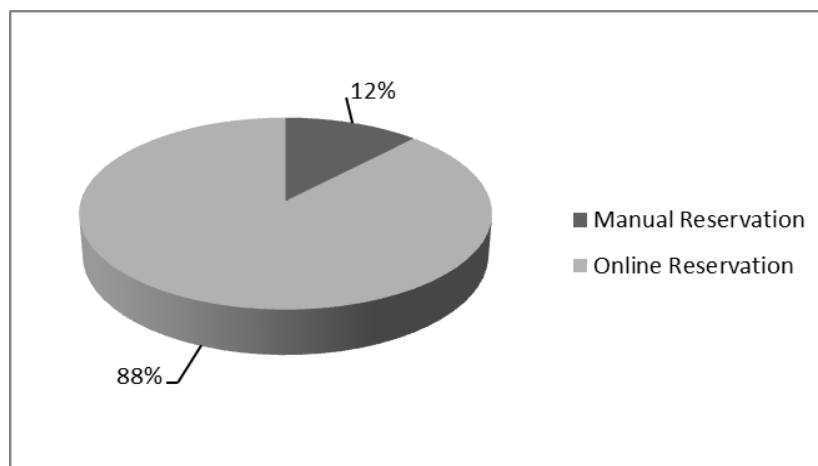


Figure 5.4: Preference between Online vs Manual System

Answering to the questions directed at the evaluators whether they would prefer the manual reservation system or the proposed online reservation system, 88% agreed to use the web based online reservation system. Demonstration of the proposed system (project background) prior to the feedback was beneficial to the evaluators when providing answers.

The users that do not prefer the online system were asked for the reasons and could receive the following responses:

- Cannot guarantee 100% as they do not meet the tour agency members
- Fear of online payment transactions
- Sometimes the websites are not up to date
- Hidden charges may apply (Less transparency)

## 5.5 User-Friendliness of menus and interfaces

The evaluators' comments for the user friendliness of the system interfaces are shown below:

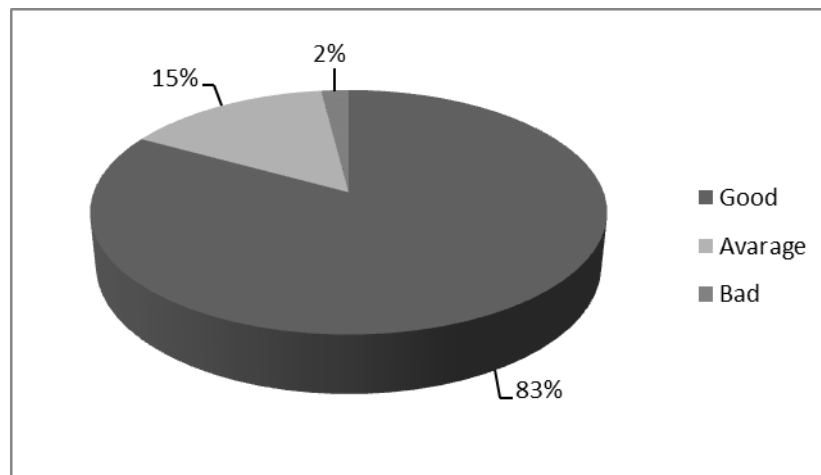


Figure 5.5: Evaluation feedback of User Interface

## 5.6 Usability of the proposed solution

Web based Package Reservation System lets its customers to place reservations and plan for their holiday spending via website. We should also note that there are a lot of existing Tour booking systems which offers only that Tour component.

The proposed solution is not confined to a specific component reservation with pre-defined set of Tours. The prototype was designed with maximum flexibility to handle many components to bundle with package. Therefore the solution offered was a practical scenario and is capable of being adapted to specific conditions encountered by real situations demanded by a necessitated project.

## 5.7 Testing

Testing of the system is carried out as a parallel activity together with system development and implementation. Each web page forms a module in the code model of this application. Therefore each page written in Php code should be tested in isolation. Testing was also carried out to ensure that the outputs of one web page are compatible and consistent with the inputs required by another related web page written in Php code. Therefore, module testing and integration are carried out during the system testing. A test plan is designed by defining the input and the output required and comparing it with the actual output. This approach is also known as the Black Box Testing technique.

## 5.8 Test Plan

The test strategy consists of a series of different tests that will fully exercise the system. The primary purpose of these tests is to uncover the system's limitations and measure its full capabilities. Following are a list of various planned tests and a brief explanation of them:

- **System Test-**

The system tests were focused on the behaviour of the system. User scenarios were executed against the system as well as screen mapping and error message testing. Overall, the system tests test the integrated system and verify that it meets the requirements defines in the requirements document.

- **Performance Test-**

Performance test was conducted to ensure that the system's response times meet the user expectation and does not exceed the specific performance criteria.

- **Security Test-**

Security tests determine how secure the system is. The tests verified that an unauthorized user access to confidential data is prevented. Allowing different users to log in and verify the available user options with the user rights.

Following format is used to measure the system output in Black Box Testing:

<b>Test Case No.</b>	01			
<b>Form</b>	Login form			
<b>Functional Specification</b>	User authentication			
<b>Test objective</b>	To check whether the entered user name and password are valid or invalid			
<b>Test data</b>	User Name: Password:			
<b>Step No.</b>	<b>Steps</b>	<b>Data</b>	<b>Expected results</b>	<b>Actual results</b>
1				
2				

Table 5.1: Test Case format



Actual test results:

<b>Test Case No.</b>	01			
<b>Form</b>	Login form			
<b>Functional Specification</b>	User Authentication			
<b>Test objective</b>	To check whether the entered User name and Password are valid or invalid			
<b>Test data</b>	User Name:  Password:			
<b>Step No.</b>	<b>Steps</b>	<b>Data</b>	<b>Expected results</b>	<b>Actual results</b>
1	Enter user name And press login button	User name = admin	Should display message “Invalid username or password ”	Display message “Invalid username or password ”
2	Enter password and press login button	Password = 123	Should display message “Invalid username or password ”	Display message “Invalid username or password ”
3	Enter username and password and then press login button	Username = admin And Password = test	Should display message “Invalid username or password ”	Display message “Invalid username or password ”
4	Enter username and password and then press login button	Username = XYZ And	Should display message “Invalid username or password ”	Display message “Invalid username or password ”

		Password = CEOS	password ”	
5	Enter username and password and then press login button	Username = XYZ And Password = 123	Should display message “Invalid username or password ”	Display message “Invalid username or password ”
6	Enter username and password and then press login button	Username = “” And Password = “”	Should display message “Invalid username or password ”	Display message “Invalid username or password ”
7	Enter username and password and then press login button	Username = admin And Password = 123	Should navigate to home page	Navigate to home page

Table 5.2: Example of Actual results format

## Acceptance Testing

This test is done prior to the release of the software to the client. The primary goal is to increase the developer’s confidence that the system meets client specific requirements. The testing will usually be done with black box testing technique and it is based on the actual data collected from the client/environment. At this level, system developer will not be participate in the testing and generally client will do this testing. According to the work schedule of the project, this testing was done by myself.

The below table shows the main tests done on the user acceptance criteria (which were collected from the community)

Acceptance Test Criteria	Status (Accept/Reject)
Only allowed system users can login to the system	<b>Accept</b>
All input data into forms are validated prior to save in the database	<b>Accept</b>

Applicable buttons, text boxes, dropdowns etc. should be available in forms	<b>Accept</b>
All the search results should give correct expected output	<b>Accept</b>
Input data into the database and retrieving data from the database should be reliable and accurate	<b>Accept</b>

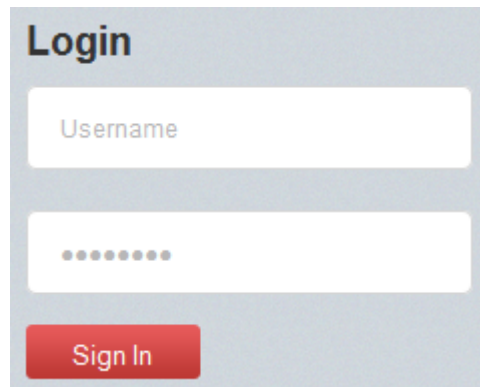
Table 5.3 : High Level Acceptance Test Criteria

Test case execution on registering new customer are as follows:

<b>No.</b>	<b>Test Case</b>	<b>Expected Output</b>	<b>Actual Output</b>	<b>Status (Pass/Fail)</b>
01	User not entered any value for 'First Name' field	Alert message ("Enter your first name") Mandatory	Alert message ("Enter your first name") Mandatory	<b>Pass</b>
02	User does not enter an value for email which is not in email format (ex. abc@gmail.com)	Alert message ("Email address not valid")	Alert message ("Email address not valid")	<b>Pass</b>
03	User enter alphabetic values for 'Telephone'	Alert message ("Please enter a valid number")	Alert message ("Please enter a valid number")	<b>Pass</b>
04	User click on 'Register account' without filling any of the mandatory fields	Alert message infront of each mandatory field to enter them	Alert message infront of each mandatory field to enter them	<b>Pass</b>

Table 5.4 : Test case execution table

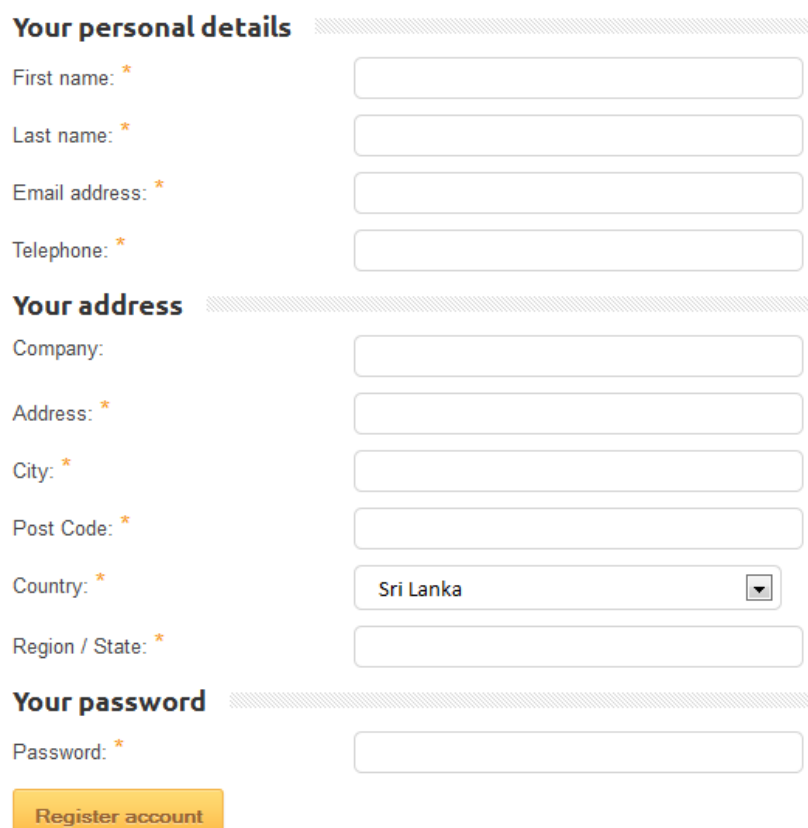
The tested validations of the customer login popup is as follows:



The image shows a login popup with a grey background. At the top left, the word "Login" is written in a bold, dark font. Below it, there are two white input fields. The first field is labeled "Username" in a light grey font. The second field is a password field, indicated by a series of eight dots. Below the password field is a red button with the text "Sign In" in white.

Figure 5.6: Login popup used for validation testing

Some of the tested validation fields for the customer registration form is as follows:



The image shows a customer registration form with three main sections, each separated by a horizontal line with a repeating pattern. The first section is titled "Your personal details" and contains four input fields: "First name: \*", "Last name: \*", "Email address: \*", and "Telephone: \*". The second section is titled "Your address" and contains six input fields: "Company:", "Address: \*", "City: \*", "Post Code: \*", "Country: \*" (with a dropdown menu showing "Sri Lanka"), and "Region / State: \*". The third section is titled "Your password" and contains one input field: "Password: \*". At the bottom of the form is a yellow button with the text "Register account".

Figure 5.7: Fields used for validations in customer registration form

## 5.9 Evaluation and analysis of feedback on project

Finally a feedback form for some of selected individuals (4 University students, 4 individuals who usually use online booking for travelling and 2 IT professionals working in similar reservation industry) were given on the following factors in a questionnaire :

- Speed and performance
- Color of the website
- User friendliness
- Navigation
- Security
- Design
- Photos
- Customer flow
- Visitor flow
- Administrator flow
- 

### Table of feedback results:

<i>Table 1:</i> Question Number	Factors	Excellent	Good	Average	Fair	Poor
1	Speed and performance	6	2	1	1	0
2	Color of the website	5	4	1	0	0
3	User friendliness	6	3	0	1	0
4	Navigation	7	2	1	0	0
5	Security	6	3	1	0	0
6	Design	6	1	2	1	0
7	Photos	10	0	0	0	0
8	Customer flow	8	1	1	0	0
9	Visitor flow	7	3	0	0	0
10	Administrator flow	9	1	0	0	0

Table 5.5: Figures of count from users'feedback

Following chart illustrates the overall picture of the feedback given by users on the developed project:

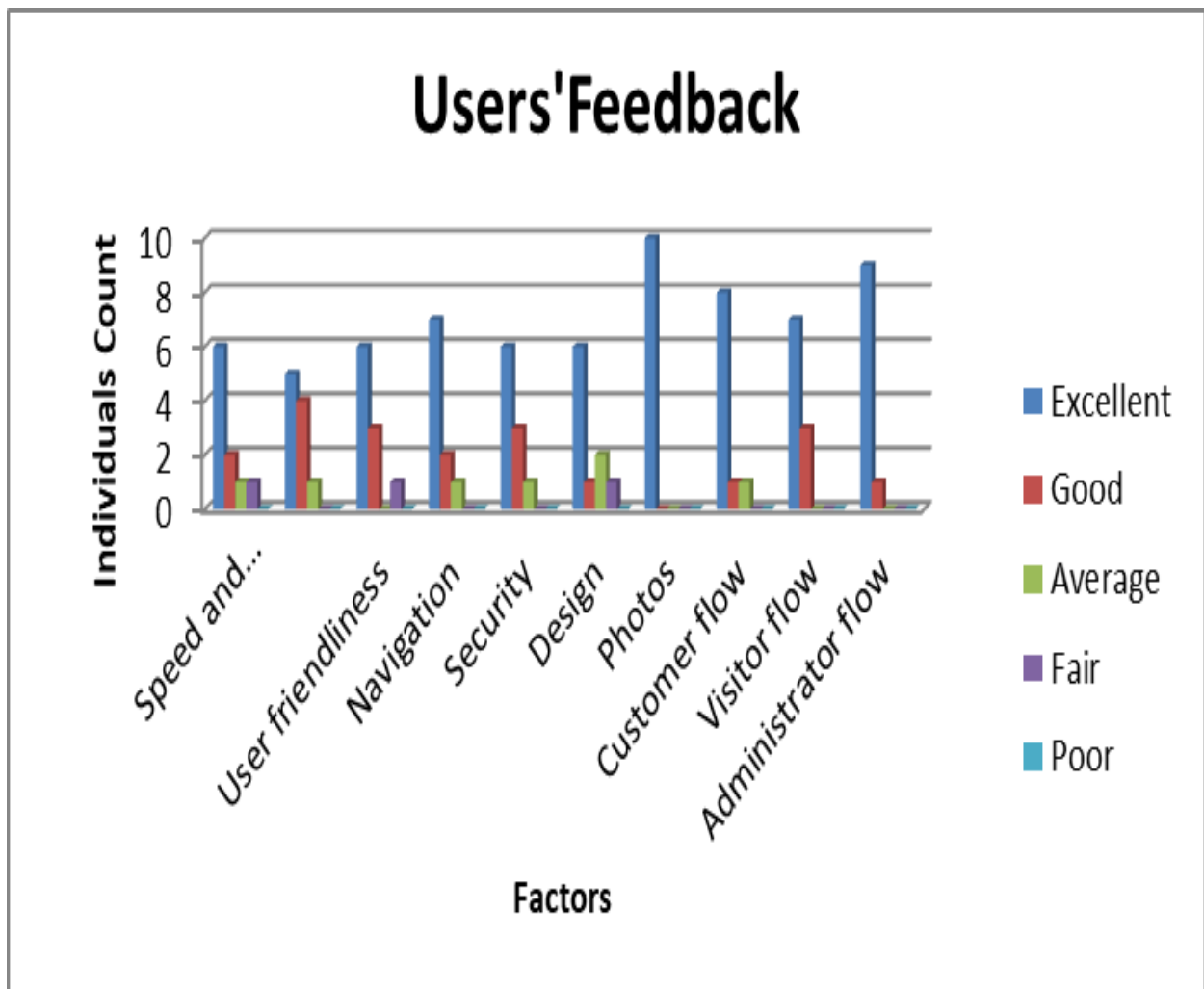


Figure 5.8: Feedback on factors analysis chart

## **5.10 Deviations from the project plan**

There were no significant deviations from the initial project plan. However there were small shifts in timeline of the expected schedule due to unavoidable circumstances. The main features of the proposed system could be delivered according to the requirements collected.

## CHAPTER 6

# CONCLUSION AND FUTURE ENHANCEMENTS

This chapter will describe the overall implementation of the project with summarized content including future enhancements. The core functionalities and the domain requirements are in the existing project it can function as expected without the future enhancements. All the enhancements planned to be added to the system. Significant work was carried out during requirement analysis, designing and implementation stages. The vision and the mission of the project discovered properly and implemented in line with the project plan.

However, designing and implementing phases were time consuming than the expected and when designing the initial plan/ there were time to time changes done for the diagrams and a huge effort had to be put when finalizing the initial footprint.

Drawing the class diagram and Use case diagrams were tricky for me as it did not exist a specific client and this project was planned to develop as a product base.

When designing the User Interface mock-ups, it had to consider the material effects of the icons, buttons etc. And it was critical to use colors for web elements in each page as they should be related to specific actions. So they were used according to each context where the each user journey navigations were divided into. This project is completely done using HTML, AJAX , JAVA SCRIPT with the support of MySQL database and php.

Finally, the project achieved it's goals as expected and there will be some future enhancements as well. Due to the time limitation those enhancements were kept aside to focus on the main business scenarios.

## 6.1 Future Enhancements

As this project is web based and relates to international tour reservation domain, this contains multiple currencies for pricing the packages. But currently it only supports to change the currency type and cannot convert the changed currency automatically according to the current exchange rates to be retrieved from externally. This can be considered as the most prioritized future enhancement as this benefits all user groups including visitor type as well.

The next enhancement will be to extend the current system in to product base purchasing as well. Then it will cater the customers to buy international products online to the door step as in 'Ebay'. So this will be planned to implement by introducing new shipping cost criteria, 3<sup>rd</sup> party integrations, Global Positioning System (GPS) etc.

Then payment section planned to be enhanced to integrate 3<sup>rd</sup> party gateway such as linking with 'PayPal'. This would be a major and a critical phase where customers as well as 3<sup>rd</sup> parties recommendation of the software product.



And the next enhancement in the priority level will be adding the discounts, coupon schemas for the customers to increase the share of the business as a marketing practice. This will benefit the suppliers as well as the customers to attract to the website.

## REFERENCES

- [1] Nirushiya Selvarathna, “Web Based Student Registration and Administration System.” University of Colombo. M.Sc. Thesis 2009/MITS1, 2009.
- [2] Hostinger Tutorials. (2019). What is Apache? An In-Depth Overview of Apache Web Server. [online] Available at: <https://www.hostinger.com/tutorials/what-is-apache> [Accessed 16 Feb. 2019].
- [3] Google Books. (2019). PHP and MySQL For Dummies. [online] Available at: [https://books.google.lk/books?id=kf8MtrLYK4C&printsec=frontcover&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.lk/books?id=kf8MtrLYK4C&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false) [Accessed 12 Mar. 2019].
- [4] Nusphere.com. (2019). PHP History. [online] Available at: [http://www.nusphere.com/php/php\\_history.htm](http://www.nusphere.com/php/php_history.htm) [Accessed 17 Apr. 2019].
- [5] JavaScript, A. (2019). An Introduction to JavaScript. [online] Javascript.info. Available at: <https://javascript.info/intro> [Accessed 17 Apr. 2019].
- [6] Anon, (2019). [online] Available at: <https://www.quora.com/What-is-the-importance-of-online-booking-systems-in-todays-era-for-any-travel-agency-business> [Accessed 4 Jan. 2019].
- [7] PHPTRAVELS. (2019). PHPTRAVELS | Travel Technology Partner. [online] Available at: <https://www.phptravels.net/> [Accessed 14 Feb. 2019].
- [8] Travelomatix.com. (2019). Travel Portal Development, B2C B2B Web Portal Design Company, White Label Portal Solutions. [online] Available at: <http://www.travelomatix.com/index.html> [Accessed 11 Jan. 2019].

# APPENDICES

## APPENDIX A – USER MANUAL

### Introduction

This section is to provide a complete user guide for the Tour Package Reservation System.

## A.1 Customer/Visitor Module

### 1. Home page

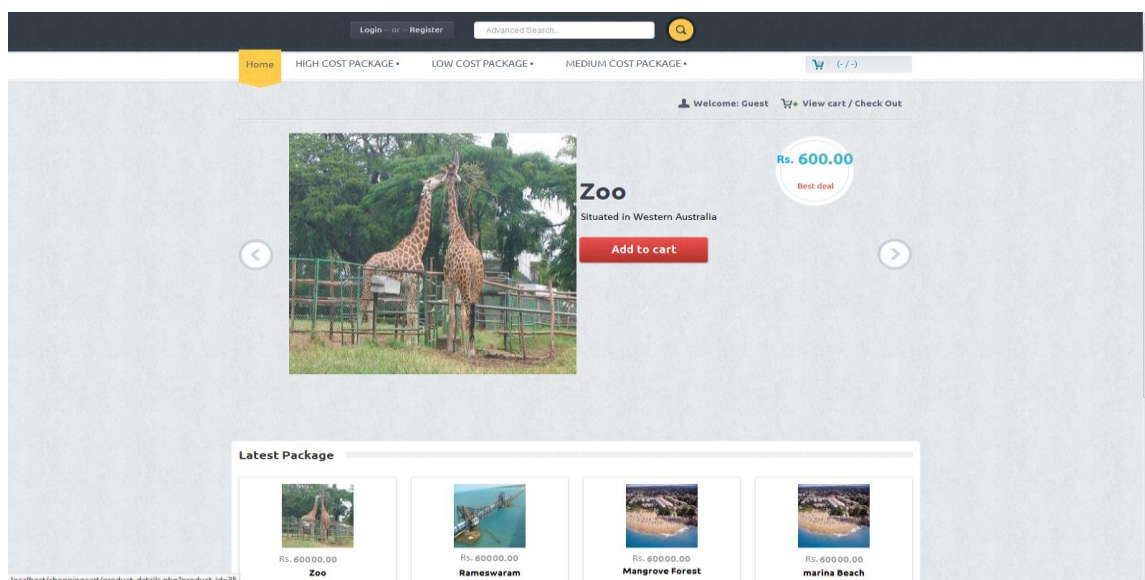


Figure A.1: Home page

This page will be the initial landing page for customer or visitor. By selecting the URL user will be navigated to this page. Here, user can perform any of the following tasks:

- Login with existing credentials for registered customers
- Visitor can newly register by initiating a registration form to be a customer
- Visitor can search and see the available products by click on each tab and displaying products. (By clicking on left and right arrows user can view detailed information on available products)

## 2 Login/ Registration

The screenshot displays a web interface for user authentication. At the top, there is a navigation bar with a 'Login or Register' button. A modal window is overlaid on the page, containing two options: 'Login' and 'Register Now'. The 'Login' section has input fields for 'Username' and 'Password', along with a 'Sign In' button and a 'Forgot password?' link. The 'Register' section includes a 'Register Now' button and a brief instruction: 'New user? By creating an account you will be able to shop faster, be up to date on an order's status...'. Below the modal, the 'Create New Account' form is visible, starting with an 'Account Type' dropdown. It is divided into three main sections: 'Your personal details' (with fields for first name, last name, email address, and telephone), 'Your address' (with fields for company, address, city, post code, and a dropdown for country set to 'Sri Lanka'), and 'Your password' (with a password field). A 'Register account' button is located at the bottom of the form. The footer of the page contains the text 'Copyright © 2018 Online Tour Packages'.

Figure A.2: Login/Registration page

1. When user click on the 'Login or Register' button, it will display a popup.
2.
  - With that popup, user will be able to directly login to the system as an existing customer with correct credentials.
  - Once user click on the 'Register Now' in the above popup in Home Page, the following 'Create New Account' form will appear.

'Create New Account' form will allow user to enter new details and validations such as required (\*) are there in the form. It will contain some mandatory fields and dropdowns as displaying in the screenshot.

### 3. View Packages

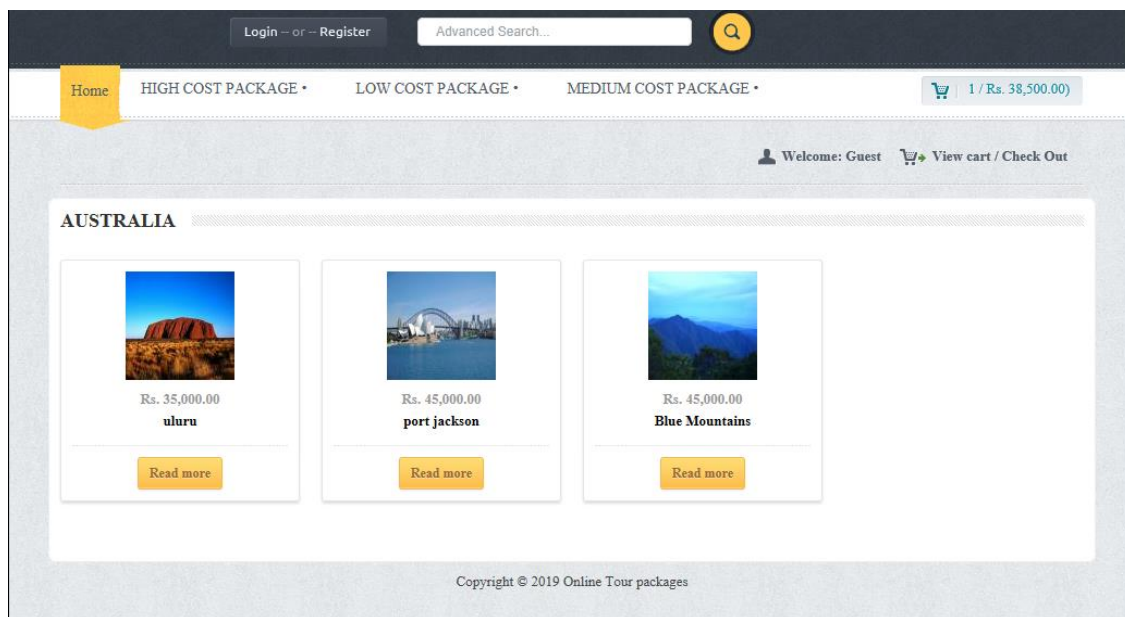


Figure A.3: View Packages page

Once visit the site by click on the tabs in home page, user will be able to view the packages. Here, user can click any of the package and navigate to view a detailed description of the package.

## 4. View User Profile

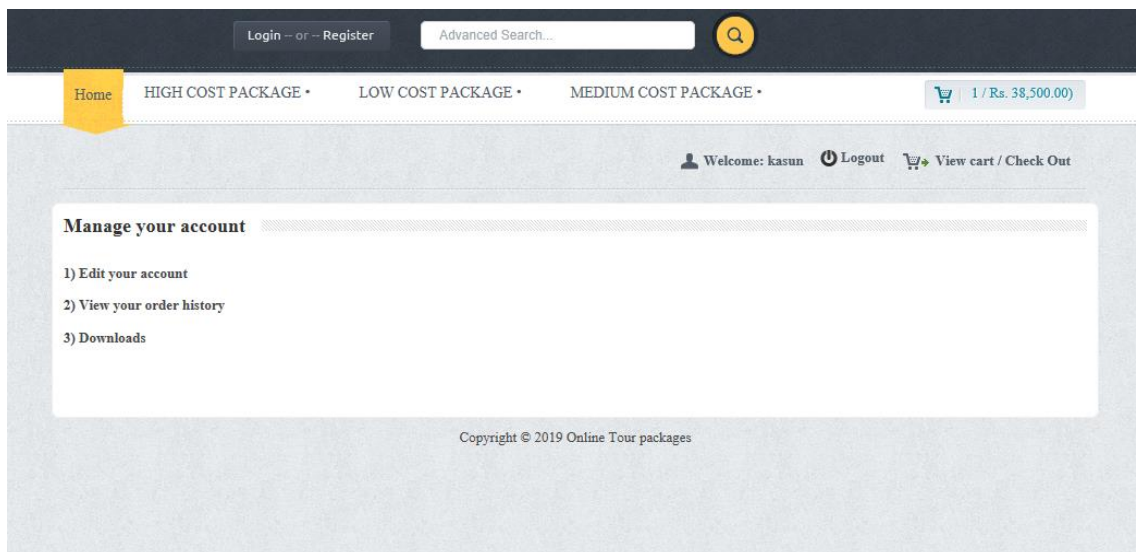


Figure A.4: View User Profile page

Once an existing customer login with correct credentials, he/she will be navigated to his/her profile. On that page user can mainly view the following available actions:

- Edit the account
- View order history
- Downloads (If available pending downloads)

Here customer will be able to view the cart if there's any existing selected packages by clicking on (**Profile detail section -> 'View cart/Check Out' button**)

Once the customer selectes any package as in the above 'View Package' section, user will then be navigated to the detailed view of the product.

## 5. Add to Cart

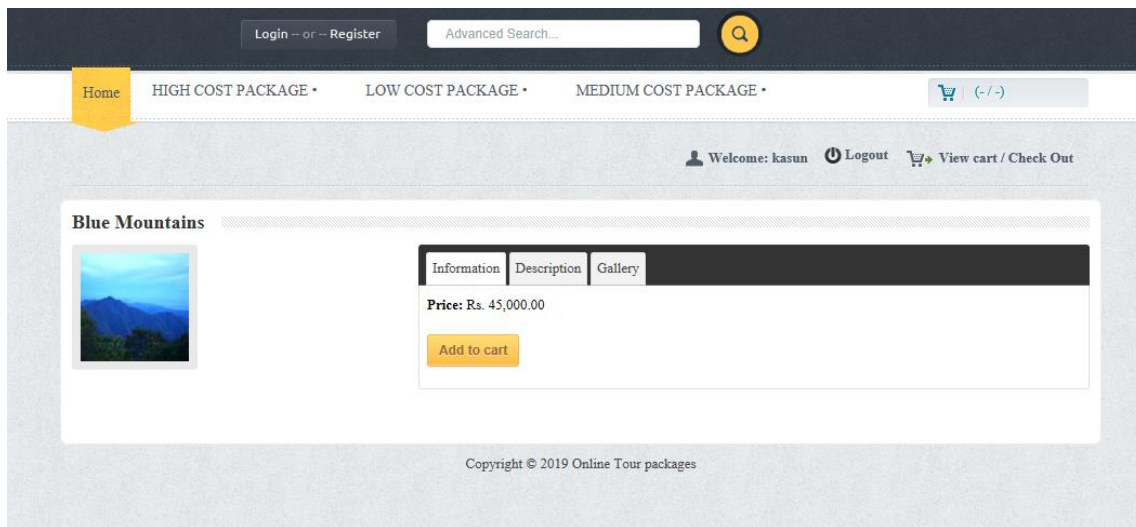


Figure A.5: Add to Cart page

It will consist with the following details (in tabs):

- Information (Price with currency, Add to Cart button)
- Description tab (To view detailed description of the tour package)
- Gallery (To view other available images of the package)

By clicking on the 'Add to cart' user can add the same package to the cart.

When user clicks on on (**Profile detail section** -> '**View cart/Check Out**' button) user will be navigated to view the cart.

## 6. View Cart

The screenshot displays the 'View Cart' page of an online tour package website. The page features a dark navigation bar at the top with 'Login - or - Register' and an 'Advanced Search...' field. Below this is a menu with 'Home', 'HIGH COST PACKAGE', 'LOW COST PACKAGE', and 'MEDIUM COST PACKAGE', along with a shopping cart icon showing '1 / Rs. 49,500.00'. The main content area shows a 'Shopping cart summary' table with columns for 'Package', 'Number of Member', 'Cost', and 'Total'. A single item, 'Blue Mountains', is listed with 1 member, a cost of Rs. 45,000.00, and a total of Rs. 45,000.00. Below the table, there are buttons for 'Update', 'Empty', and 'Checkout'. The page also includes a 'Welcome: kasun' message, a 'Logout' button, and a 'View cart / Check Out' link. The footer contains the copyright notice 'Copyright © 2019 Online Tour packages'.

Package	Number of Member	Cost	Total
Blue Mountains	<input type="text" value="1"/>	Rs. 45,000.00	Rs. 45,000.00
		<b>Sub-Total:</b>	Rs. 45,000.00
		<b>Tax (VAT 10 %):</b>	Rs. 4,500.00
		<b>Total:</b>	Rs. 49,500.00

Update   Empty   Checkout

Copyright © 2019 Online Tour packages

Figure A.6: View Cart page

This will be the view cart page and here user will be able to do the following tasks:

- Update cart (increasing number of members to the tour)
- View any tax available (With “Sub-Total” and “Total” fields)
- Empty the cart (Click on the ‘X’ mark or ‘Empty’ button)
- Navigate to the payment page (Click on ‘Checkout’ button)



## 7. Payment page (Checkout)

The screenshot shows a checkout page with the following elements:

- Shipping address:** kasun, perera, 23, orlando, 1234, Sri Lanka, western
- Payment address:** kasun, perera, 23, orlando, 1234, Sri Lanka, western
- Payment method:**
  - PayPal (radio button)
  - Bank transfer (radio button) with input fields for Bank Name \*, Account No \*, and password \*
  - Cash on delivery (radio button)
- Add comment about your order:** A large text area for user comments.
- Confirm order:** A yellow button to proceed with the order.
- Copyright © 2019 Online Tour packages** (small text at the bottom)

Figure A.7: Payment/Checkout page

Once user navigate to the 'Checkout' page from the 'View Cart' page, it will display the following fields for the user proceed the payment (Confirm the reservation):

- Display the Payment address and the Shipping address (Shipping address was introduced mainly to extend the website for product selling)
- Payment methods (Paypal (Not integrated payment gateway), Bank transfer, Cash on delivery – selectable with radio buttons)
- Text box (To add any comments regarding the order)
- 'Confirm Order' button (To confirm the order)

## 8. Order Confirmation page

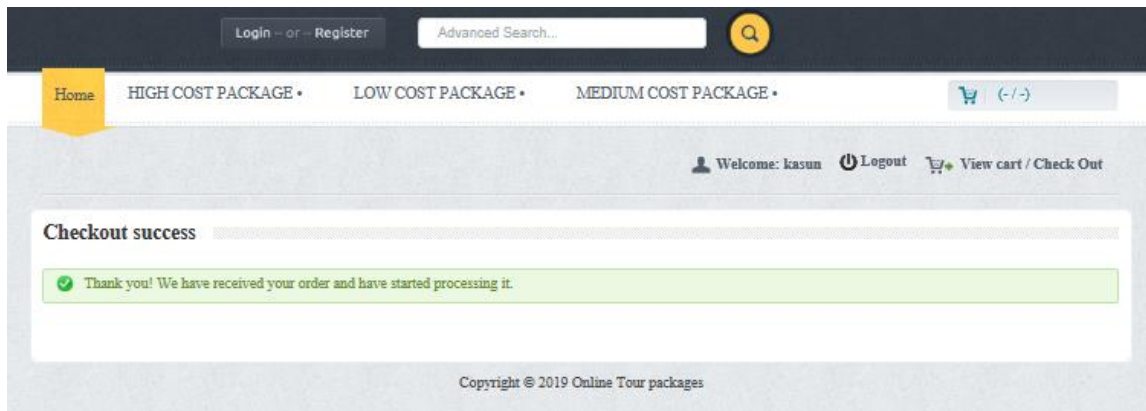


Figure A.8: Order Confirmation page

Finally when user fill the details in the payment page and click on the 'Checkout' button, system will navigate the user to the order confirmation page and there customer can view the order confirmation message.

## A.2 Administrator Module

### 1. Login page

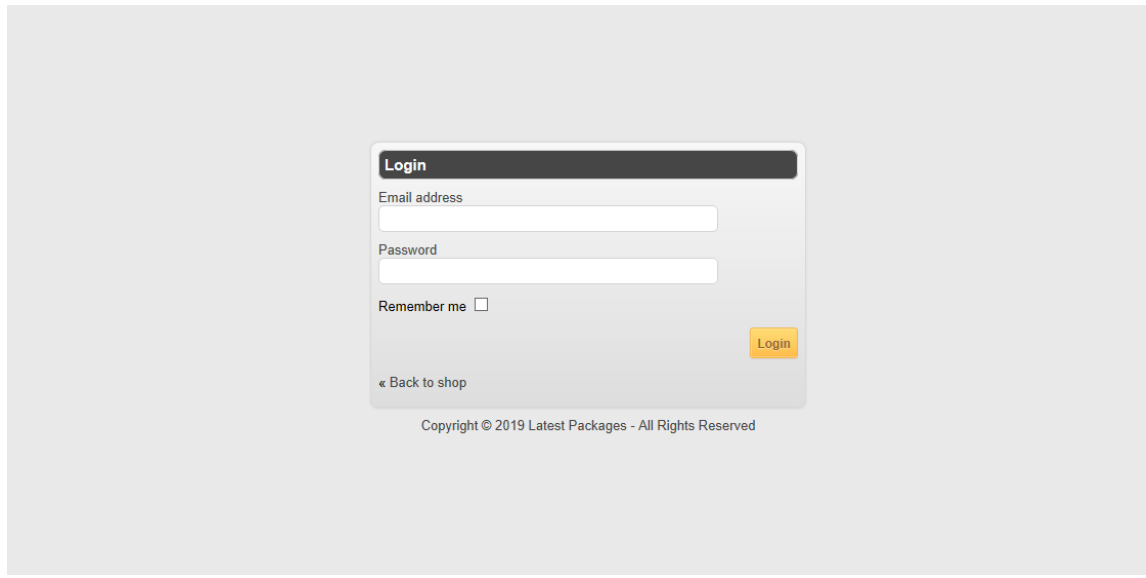


Figure A.9: Administrator Login page

This page will be the initial login page for administrator. By selecting the URL available for administrator, he/she will be navigated to this page. Here, admin can enter the correct credentials and log in to the administrator profile.

## 2. DASHBOARD tab

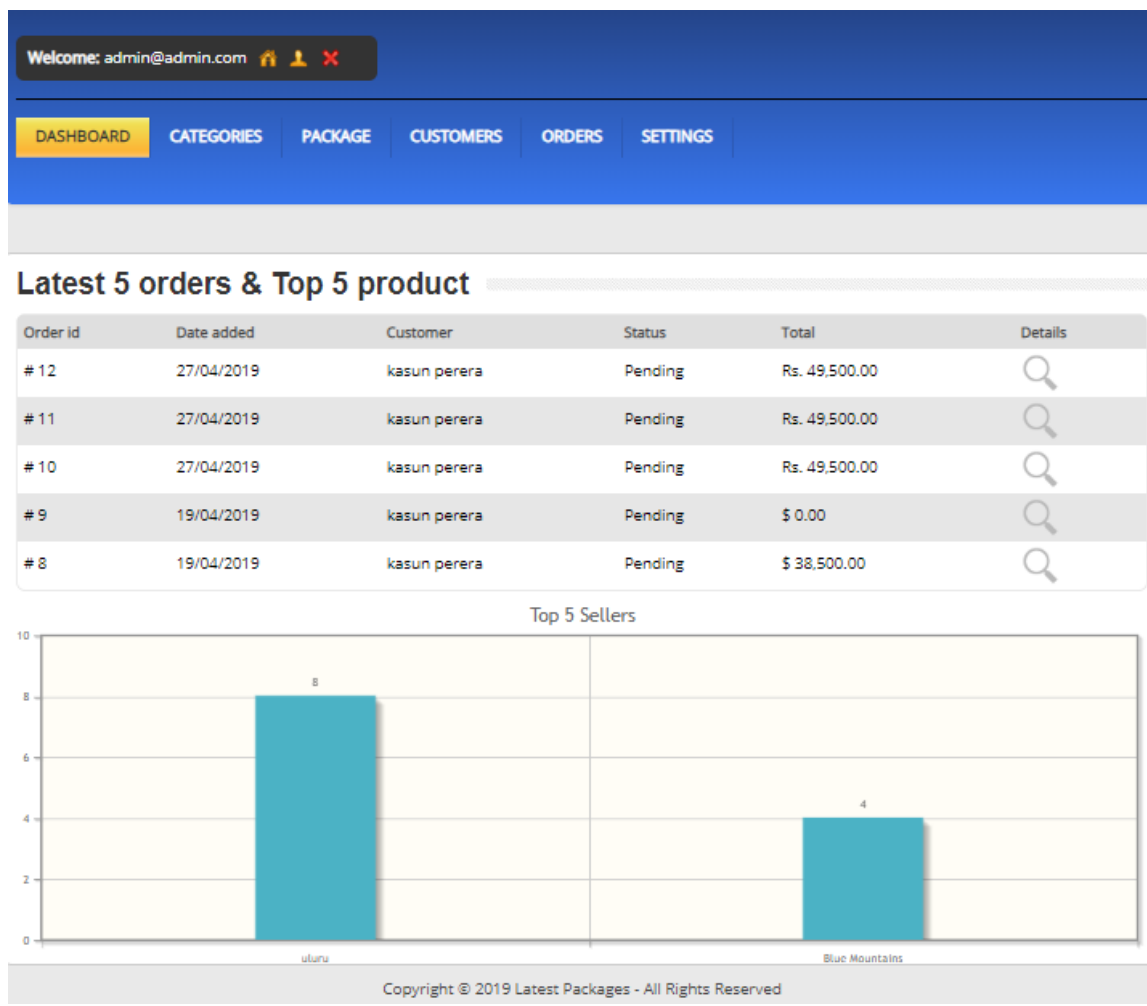


Figure A.10: Dashboard tab (administrator module)

Once administrator login with correct credentials, system will navigate to this dashboard. This will contain the following elements:

- Details of latest 5 orders (Includes Order id, Date added, Customer, Status, Total)
- Details column in front of each order (To view detailed description of the specific order)
- Statistics on top five sellers according to the orders done (Totals with bar chart)

### 3. CATEGORIES tab

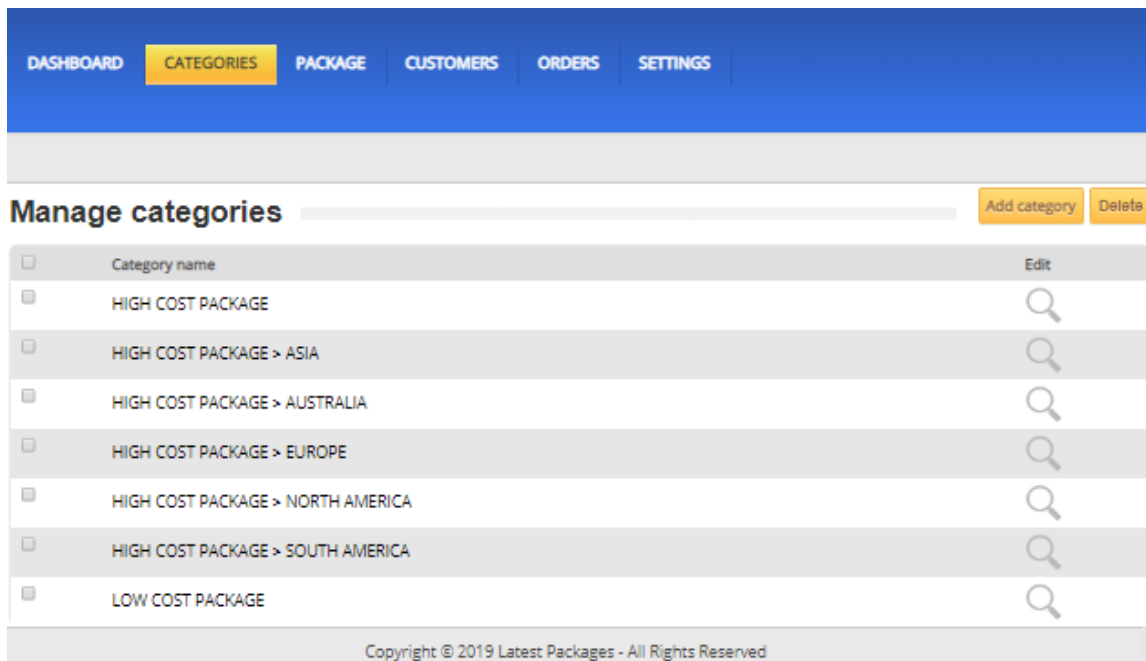


Figure A.11: Categories tab (administrator module)

In CATEGORIES tab, user can do the following tasks:

- Delete existing categories (Multiple deletion by check boxes are available)
- Edit an existing category (There administrator is able to Change the Name, Parent Category and Category status)
- Add new category (Able to add Category Name, Parent Category and Category status)

## 4. PACKAGE tab

Welcome: admin@admin.com

DASHBOARD CATEGORIES **PACKAGE** CUSTOMERS ORDERS SETTINGS

**Manage Package** Add Package Delete

Filter by name Filter by Cost Filter by member Filter

<input type="checkbox"/>	Preview image	Package name	Cost	Num of member	Action
<input type="checkbox"/>		Zoo	Rs. 600.00	39	
<input type="checkbox"/>		marina Beach	Rs. 500.00	1	
<input type="checkbox"/>		port jackson	Rs. 45,000.00	0	
<input type="checkbox"/>		Blue Mountains	Rs. 45,000.00	3	

Next page »

Copyright © 2019 Latest Packages - All Rights Reserved

Figure A.12: Package tab (administrator module)

In PACKAGE tab,

administrator can do the following tasks:

- Delete exiting packages (Multiple deletion by check boxes are available)
- Edit an existing package (There administrator is able to Change the Category related, Name of package, Description, Image, set Price, Quantity available, Product expiry etc )
- Add new package (There administrator is able to Change the Category related, Name of package, Description, Image, set Price, Quantity available, Product expiry etc )
- Filtering option (By Name, Cost or Member)
- Pagination (By 'Next Page' button)

## 5. CUSTOMERS tab

Manage customers

Add customer Visitor Track List Delete

Filter by name Filter by email Filter

<input type="checkbox"/>	Client name	Email address	Registered since	Action
<input type="checkbox"/>	kasun perera	kasun@gmail.com	18/04/2019 13.43.39	

Figure A.13: Customers tab (administrator module)

In CUSTOMERS tab,

an administrator is able to perform the below actions:

- View customer details in a table (Contain colums: Client name, Emain address, Registered date, Action(Edit))
- Filter the above results (By Customer Name, Email)
- Add a new customer (Here administrator will be navigated to a new customer registration form same as in customer/visitor flow)
- Visitor Track List (To display the recently visited persons to view the web site-contains Date of visit, Time, Month, Year, administrator will only be able to delete an entry)
- Delete exiting customer (Multiple deletion by check boxes are available)
- Edit an existing customer (Here administrator will be navigated to existing registration detail form same as in customer/visitor flow)

## 6. ORDERS tab

Welcome: admin@admin.com

DASHBOARD CATEGORIES PACKAGE CUSTOMERS **ORDERS** SETTINGS

### Manage orders

Filter by order id  Filter by customer  Filter by status All ▼ Filter

Order id	Customer	Status	Total	Action
# 12	kasun perera	Pending	Rs. 49,500.00	
# 11	kasun perera	Pending	Rs. 49,500.00	
# 10	kasun perera	Pending	Rs. 49,500.00	
# 9	kasun perera	Pending	\$ 0.00	
# 8	kasun perera	Pending	\$ 38,500.00	
# 7	kasun perera	Pending	\$ 38,500.00	

Next page »

Copyright © 2019 Latest Packages - All Rights Reserved

Figure A.14: Orders tab (administrator module)

In ORDERS tab,  
an administrator is able to perform the below actions:

- View the existing order details (By 'Order id', 'Customer', 'Status', 'Total' columns in the table)
  - Edit existing orders by customers:  
In edit orders page, administrator will be able to view/perfrom below tasks:
    - View Payment address, Payment method, Product name, Number of packages, Price, Total, Tax (if applicable), Date added, Current order status)
    - Administrator is able to change the Order status by a dropdown as (Pending, processing, Cancelled, Failed etc.)
    - Once click on 'Add status' button, it will save the changed status.
  - Filter the existing orders (By 'order id', 'customer', 'order status')
- 'Next page' button for pagination.



## 7. SETTINGS tab

Welcome: admin@admin.com

DASHBOARD CATEGORIES PACKAGE CUSTOMERS ORDERS **SETTINGS**

### Settings

General Cart Upload

Currency: Sri Lankan Rupee

Currency position: Left

Shipping cost: 0

Tax description: VAT

Tax rate: 10  Apply to shipping

Save

Figure A.15: Settings tab (administrator module)

In SETTINGS tab,

Administrator can perform the following actions:

- Change the currency type
- Change the position of currency displayed
- Tax description to be displayed in the View cart page
- Applicable Tax rate amount
- 'SAVE' button to save all changes
- Upload any images for background

## APPENDIX B – SOURCE CODES

### 1. PHP source code to View Account

```

<?php

//Include the common file
require_once('common.php');

//Check if the user is logged in
if (!$authentication->logged_in() || !$authentication->is_group('customer')) header("Location:
login.php");

require_once('Connections/shoppingcart.php');

$msg=0;
$m=mm=date("m");
$y=yy=date("Y");
mysql_select_db($database_shoppingcart, $shoppingcart);
$query_valchk ="SELECT * FROM orders WHERE user_id =".$_SESSION['user_id']."' and
ormonth='".$m.'" and oryear='".$y.'" ORDER BY date_added DESC";
$valchk = mysql_query($query_valchk, $shoppingcart) or die(mysql_error());
$row_valchk = mysql_fetch_assoc($valchk);
$totalRows_valchk = mysql_num_rows($valchk);

mysql_select_db($database_shoppingcart, $shoppingcart);
$query_valcup ="SELECT * FROM coupons WHERE coupon_status='1' ORDER BY
coupon_id DESC";
$valcup = mysql_query($query_valcup, $shoppingcart) or die(mysql_error());
$row_valcup = mysql_fetch_assoc($valcup);
$totalRows_valcup = mysql_num_rows($valcup);

if($totalRows_valchk == 5 && $totalRows_valchk <=6)
{
setcookie("msg",'1',time()+(100*24));
setcookie("cupon",$row_valcup['coupon_code'],time()+(100*24));
setcookie("edate",$row_valcup['date_end'],time()+(100*24));

$updateSQL      ="UPDATE      `users`      SET      couponsts='1'      WHERE
user_id='".$_SESSION['user_id']."'";
mysql_select_db($database_shoppingcart, $shoppingcart);
$result1 = mysql_query($updateSQL, $shoppingcart) or die(mysql_error());

}

```

```
//Display the template
$tpl->display('account');
```

```
?>
```

## 2. PHP source code to Create account

```
<?php if (__FILE__ == $_SERVER['SCRIPT_FILENAME']) exit('No direct access
allowed.');
```

```
<?php if (count($cart->get_cart()) > 0): ?>
```

```
    <?php
```

```
        $quantity = 0;
```

```
        foreach ($cart->get_cart() as $value)
            $quantity += $value['product_quantity'];
```

```
    ?>
```

```
    <p><?php echo $quantity; ?> / <?php echo price($cart->total()); ?></p>
```

```
<?php else: ?>
```

```
    <p>(- / -)</p>
```

```
<?php endif; ?>
```

## 3. PHP source code to Display Cart

```
<?php if (__FILE__ == $_SERVER['SCRIPT_FILENAME']) exit('No direct access
allowed.');
```

```
<?php if (count($cart->get_cart()) > 0): ?>
```

```
    <?php
```

```
        $quantity = 0;
```

```
        foreach ($cart->get_cart() as $value)
            $quantity += $value['product_quantity'];
```

```
    ?>
```

```
<p><?php echo $quantity; ?> / <?php echo price($cart->total()); ?></p>
```

```
<?php else: ?>
    <p>(- / -)</p>
<?php endif; ?>
```

## 4. PHP source code for index.php

```
<?php

//Include the common file
require_once('common.php');
include("usertrack.php");
//Products

$products = array();

foreach ($db->query("SELECT * FROM " . config_item('cart', 'table_products') . " ORDER
BY product_id DESC LIMIT " . config_item('cart', 'per_page_catalog') . "")) as $row) {

    $products[] = array(
        'product_id'           => $row['product_id'],
        'product_thumbnail'    => $row['product_thumbnail'],
        'product_price'        => $row['product_price'],
        'product_name'         => $row['product_name'],
        'product_description'   => $row['product_description'],
        'product_image'        => $row['product_image']
    );
}

//Template values
$tpl->set('products', $products);

//Display the template
$tpl->display('index');

?>
```

## 5. PHP source code for login.php

```
<?php

//Include the common file
```

```

require_once('common.php');

//Check if the user is logged in
if ($authentication->logged_in() && $authentication->is_group('customer'))
header("Location: account.php");

//Check if the form has been submitted
if (isset($_POST['login'])) {

    $validate->email($_POST['email'], 'Email address not valid. ');
    $validate->required($_POST['password'], 'Enter your password. ');

    if (!$error->has_errors()) {

        $remember = false;

        if (isset($_POST['remember']))
            $remember = true;

        if ($authentication->login($_POST['email'], $_POST['password'],
$remember))
            header("Location: account.php");
        else
            $tpl->set('failed', true);

    } else {

        $tpl->set('failed', true);

    }

}

//Check if the form has been submitted
if (isset($_POST['reset_password'])) {

    $validate->email($_POST['email'], 'Email address not valid. ');

    if (!$error->has_errors()) {

        if ($authentication->new_password($_POST['email']))
            $tpl->set('success', true);
        else
            $tpl->set('failed', true);

    }

}

```

```
//Display the template
$tpl->display('login');
```

```
?>
```

## 6. PHP source code for User track (to view the users visiting the website)

```
<?php
require_once('Connections/shoppingcart.php');
if (!function_exists("GetSQLValueString")) {
function    GetSQLValueString($theValue,    $theType,    $theDefinedValue    =    "",
$theNotDefinedValue = "")
{
    if (PHP_VERSION < 6) {
        $theValue = get_magic_quotes_gpc() ? stripslashes($theValue) : $theValue;
    }

    $theValue    =    function_exists("mysql_real_escape_string")    ?
mysql_real_escape_string($theValue) : mysql_escape_string($theValue);

    switch ($theType) {
        case "text":
            $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
            break;
        case "long":
        case "int":
            $theValue = ($theValue != "") ? intval($theValue) : "NULL";
            break;
        case "double":
            $theValue = ($theValue != "") ? doubleval($theValue) : "NULL";
            break;
        case "date":
            $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
            break;
        case "defined":
            $theValue = ($theValue != "") ? $theDefinedValue : $theNotDefinedValue;
            break;
    }
    return $theValue;
}

$timezone = new DateTimeZone("Asia/Colombo" );
$date = new DateTime();
```

```

$date->setTimezone($timezone );
    $vdate = $date->format( 'd/m/Y' );
    $vtime = $date->format( 'H:i:s A' );
    $vmonth = $date->format( 'M' );
    $vyear = $date->format( 'Y' );

mysql_select_db($database_shoppingcart, $shoppingcart);
$query_userinfo = "SELECT * FROM visitor_track where vdate='".$vdate.'" and
userip='".$_SERVER['REMOTE_ADDR']."' ORDER BY sno DESC";
$userinfo = mysql_query($query_userinfo, $shoppingcart) or die(mysql_error());
$row_userinfo = mysql_fetch_assoc($userinfo);
$totalRows_userinfo = mysql_num_rows($userinfo);

    if($totalRows_userinfo>0)
    {

        $updatesql = "update visitor_track SET vtime='".$vtime.'" where vdate='".$vdate.'" and
userip='".$_SERVER['REMOTE_ADDR']."'";
        mysql_select_db($database_shoppingcart, $shoppingcart);
        $Result12 = mysql_query($updatesql, $shoppingcart) or die(mysql_error());

    }
    else{
        $insertSQL = sprintf("INSERT INTO visitor_track (userip, vdate, vtime, vmonth,
vyear) VALUES (%s, %s, %s, %s, %s)",
            GetSQLValueString($_SERVER['REMOTE_ADDR'], "text"),
            GetSQLValueString($vdate, "text"),
            GetSQLValueString($vtime, "text"),
            GetSQLValueString($vmonth, "text"),
            GetSQLValueString($vyear, "text"));

        mysql_select_db($database_shoppingcart, $shoppingcart);
        $Result1 = mysql_query($insertSQL, $shoppingcart) or die(mysql_error());
    }
?>

```

## 7. PHP source code to view order details by administrator

```

<?php

//Include the common file
require('./common.php');

//Check if the user is logged in

```

```

if (!$authentication->logged_in() || !$authentication->is_admin()) header("Location:
login.php");

//Returns the number of rows
$row_count = $db->row_count("SELECT order_id FROM " . config_item('cart',
'table_orders'));

//Pagination
if (isset($_GET['page']))
    $current_page = trim($_GET['page']);
else
    $current_page = 1;

$start = ($current_page - 1) * config_item('cart', 'per_page_admin');

$sql = "SELECT * FROM " . config_item('cart', 'table_orders') . " o, " . config_item('cart',
'table_order_status_descriptions') . " osd WHERE o.order_status_description_id =
osd.order_status_description_id";

$implode = array();

if (isset($_POST['filter']) && !empty($_POST['filter_order_id']))
    $implode[] = " o.order_id = " . $_POST['filter_order_id'] . " ";

if (isset($_POST['filter']) && !empty($_POST['filter_name']))
    $implode[] = " CONCAT(o.first_name, ' ', o.last_name) LIKE '%" .
$_POST['filter_name'] . "%' ";

if (isset($_POST['filter']) && !empty($_POST['order_status_description_id']))
    $implode[] = " o.order_status_description_id = " .
$_POST['order_status_description_id'] . " ";

if ($implode)
    $sql .= " AND " . implode(" AND ", $implode);

$sql .= " ORDER BY o.order_id DESC";

$sql .= " LIMIT " . $start . ", " . config_item('cart', 'per_page_admin') . " ";

$pages = ceil($db->row_count("SELECT order_id FROM " . config_item('cart',
'table_orders') . " ") / config_item('cart', 'per_page_admin'));

//Orders
$orders = array();

foreach ($db->query($sql) as $row) {

    $orders[] = array(

```



```
        'order_id'          => $row['order_id'],
        'first_name'       => $row['first_name'],
        'last_name'        => $row['last_name'],
        'status_name'     => $row['status_name'],
        'currency'         => $row['currency'],
        'total'            => $row['total']
    );
}

//Template values
$tpl->set('row_count', $row_count);
$tpl->set('current_page', $current_page);
$tpl->set('pages', $pages);
$tpl->set('orders', $orders);

//Display the template
$tpl->display('admin/orders');

?>
```