UNITY UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE



PROJECT TITILE: TOUR AND TRAVEL GUDING SYSTEM FOR DESSIE TOUR GUIDE AGENCY

BY

GROUP MEMBER

1. DESALEGN ABEBE
2. ETSEGENET GEBEYAW
3. SEMAGN MEKONNEN
4. SELESHI MEKONNEN
5. TADIYOS HAILU
6. WASU NEGA

A PROJECT SUMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE OF UNITY UNIVERSITY IN PARTIAL FULFILLMENT FOR THE REQUIREMENT FOR THE DEGREE OF BACHLOR OF SCIENCE IN COMPUTER SCIENCE

ADVISOR: DANIEL MEHARI

DECEMBER, 2017

UNITY UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

PROJECT TITILE: TOUR AND TRAVEL GUDING SYSTEM FOR DESSIE TOUR GUIDE AGENCY

BY

GROUP MEMBER

1. DESALEGN ABEBE
2. TSEGENET GEBEYAW
3. SEMAGN MEKONNEN
4. SELESHI MEKONNEN
5. TADIYOS HAILU
6. WASU NEGA

Name and signature of member of examining board

**Name Title Signature Date**

Mr. Daniel Mehari (Advisor, ) \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_

Examiner, \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_

Examiner, \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_

**DECLARATION**

We declare that this project is our original work and has not been presented for a degree in any other university.

**Name Signature Date**

DESALEGN ABEBE \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

TSEGENET GEBEYAW \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

SEMAGN MEKONNEN \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

SELESHI MEKONNEN \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

TADIYOS HAILU \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

WASU NEGA \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

This project has been submitted for examination with my approval as university advisor.

**Name Signature Date**

Daniel Mehari \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_

## 

## ****Acknowledgement****

We are thankful to our advisor, Mr. Daniel Mehari, whose encouragement, guidance and support from the initial to the final level enabled us to develop an understanding of the project and complete it on time. We also like to thank, Dessie culture and tourism and their employees for their help and cooperation during our interview. Lastly, we offer our regards and blessings to all of those who supported us in any respect during the completion of the project.

content pages

[Acknowledgement i](#_Toc422269306)

[content ii](#_Toc422269307)

[List Of Tables iv](#_Toc422269308)

[List Of Figures v](#_Toc422269309)

[Abstract vi](#_Toc422269310)

[Keyword vii](#_Toc422269311)

[Abbreviation viii](#_Toc422269312)

[CHAPTER ONE 1](#_Toc422269313)

[1 System Requirement Specification 1](#_Toc422269314)

[1.1 Introduction 1](#_Toc422269315)

[*1.1.1 Background of The Organization* 1](#_Toc422269316)

[*1.1.2 Introduction about the Project* 1](#_Toc422269317)

[1.2 Statement of the problem 2](#_Toc422269318)

[1.3 Significance of the system 2](#_Toc422269319)

[1.4 Objective of the Project 3](#_Toc422269320)

[1.4 .1 General objective 3](#_Toc422269321)

[1.4.2 Specific objective 3](#_Toc422269322)

[1.5 Scope and limitation of the project 3](#_Toc422269323)

[*1.5.1 Scope* 3](#_Toc422269324)

[*1.5.2 Limitation* 4](#_Toc422269325)

[1.6 Methodology 4](#_Toc422269326)

[*1.6.1 Data Gathering* 4](#_Toc422269327)

[*1.6.2 Design methodology* 5](#_Toc422269328)

[*1.6.3 Analysis methodology* 5](#_Toc422269329)

[*1.6.4 Hardware and software used for Implementation methodology* 5](#_Toc422269330)

[*1.6.5 Test methodology* 7](#_Toc422269331)

[CHAPTER TWO 8](#_Toc422269332)

[2 Requirement Elicitations 8](#_Toc422269333)

[2.1 The over view of the existing system 8](#_Toc422269334)

[*2.1.1 Problem of the existing system* 8](#_Toc422269335)

[*2.1.2 Weakness and strength of the existing system* 9](#_Toc422269336)

[*2.1.3 Business rules of the system* 9](#_Toc422269337)

[2.2 overview of the proposed system 10](#_Toc422269338)

[*2.2.1Functional requirement of the proposed system* 10](#_Toc422269339)

[*2.2.2 Nonfunctional requirement of the proposed system* 11](#_Toc422269340)

[*2.2.3 System requirement of the proposed system* 12](#_Toc422269341)

[2.3 constraint and assumption 13](#_Toc422269342)

[*2.3.1 Constraints* 13](#_Toc422269343)

[*2.3.2 Assumptions* 14](#_Toc422269344)

[CHAPTER THREE 15](#_Toc422269345)

[3 System Modeling 15](#_Toc422269346)

[3.1 Actor description 15](#_Toc422269347)

[3.2 Use Case Identification 15](#_Toc422269348)

[*3.2.1 Use case model* 16](#_Toc422269349)

[*3.2.2 Use case description* 17](#_Toc422269350)

[3.3 sequence diagram 52](#_Toc422269351)

[3.4 Class diagram 70](#_Toc422269352)

[CHAPTER FOUR 71](#_Toc422269353)

[4 System Design 71](#_Toc422269354)

[4.1 Design Goal 72](#_Toc422269355)

[4.2 System Decomposition and Interrelation between them 73](#_Toc422269356)

[4.4 Deployment Diagram 75](#_Toc422269357)

[4.5 persistence data Management 76](#_Toc422269358)

[4.6 Access Control and Security 76](#_Toc422269359)

[4.7 User Interface Design 77](#_Toc422269360)

[Finger 26: user interfaces operator pages 79](#_Toc422269361)

[CHAPTER FIVE 80](#_Toc422269362)

[5 Implementation and Testing 80](#_Toc422269363)

[5.1 Implementation Overview 80](#_Toc422269364)

[5.1.1 Objective of the implementation 80](#_Toc422269365)

[5.1.2 Constraint of Implementation 80](#_Toc422269366)

[5.1.3 Tools and Environment 80](#_Toc422269367)

[5.1.3.1. Hardware components 81](#_Toc422269368)

[5.1.3.2 Software Components 81](#_Toc422269369)

[5.1.3.3 Environments 81](#_Toc422269370)

[5.2 Sample code and description 82](#_Toc422269371)

[Source code of login Admin 82](#_Toc422269372)

[5.3 Testing Overview 85](#_Toc422269373)

[5.3.1 Test by scope 85](#_Toc422269374)

[5.3.1.1Unit test 85](#_Toc422269375)

[5.3.1.2 Integration Test 86](#_Toc422269376)

[5.3.2 Testing by requirements 86](#_Toc422269377)

[5.3.3 Test Implementation (Validation testing) 87](#_Toc422269378)

[CHAPTER SIX 88](#_Toc422269379)

[6 User Manual 88](#_Toc422269380)

[6.1 Installation guide 88](#_Toc422269381)

[CHAPTER SEVEN 90](#_Toc422269382)

[7. Conclusion and Recommendation 90](#_Toc422269383)

[7.1 Conclusion 90](#_Toc422269384)

[7.2 Recommendation 91](#_Toc422269385)

[Reference 92](#_Toc422269386)

## List of Tables

Tables’ pages

[Table 1: Actor description 15](#_Toc422269406)

[Table 2: use case description for add profile 18](#_Toc422269407)

[Table 3: use case description for registration 20](#_Toc422269408)

[Table 4: use case description for login 22](#_Toc422269409)

[Table 5: use case description for add car 23](#_Toc422269410)

[Table 6: use case description for update car 25](#_Toc422269411)

[Table 7: use case description for delete car 27](#_Toc422269412)

[Table 8: use case description for manage tourist site 29](#_Toc422269413)

[Table 9: use case description for assign operator 31](#_Toc422269414)

[Table 10: use case description for car reservation. 33](#_Toc422269415)

[Table 11: use case description for hotel reservation 35](#_Toc422269416)

[Table 12: use case description for tourist site reservation 37](#_Toc422269417)

[Table 13: use case description for update schedules 40](#_Toc422269418)

[Table 14: use case description view hotel reservation 42](#_Toc422269419)

[Table 15: use case description for view tourist site reservation 44](#_Toc422269420)

[Table 16: use case description for comment 46](#_Toc422269421)

[Table 17: use case description for account create 48](#_Toc422269422)

[Table 18: use case description for delete account 50](#_Toc422269423)

[Table 19: use case description for generate report 52](#_Toc422269424)

## List of Figures

Figure pages

[Figure 1: use case diagram 16](#_Toc422269467)

[Figure 2: sequence diagram for add profile 53](#_Toc422269468)

[Figure 3: sequence diagram for registration 53](#_Toc422269469)

[Figure 4: sequence diagram to login 54](#_Toc422269470)

[Figure 5: sequence diagram for add car 55](#_Toc422269471)

[Figure 6: sequence diagram for update car 56](#_Toc422269472)

[Figure 7: sequence diagram for delete car 57](#_Toc422269473)

[Figure 8: sequence diagram for manage tourist site 58](#_Toc422269474)

[Figure 10: sequence diagram for car reservation 60](#_Toc422269475)

[Figure 11: sequence diagram for hotel reservation 61](#_Toc422269476)

[Figure 12: sequence diagram to tourist site reservation 62](#_Toc422269477)

[Finger 13: sequence diagram for update schedule 63](#_Toc422269478)

[Figure 14: sequence diagram for view hotel reservation 64](#_Toc422269479)

[Figure 15: sequence diagram for view tourist site reservation 65](#_Toc422269480)

[Figure 16: sequence diagram for comment 66](#_Toc422269481)

[Finger 17: sequence diagram for create operator account 67](#_Toc422269482)

[Finger 18: sequence diagram for delete operator account 68](#_Toc422269483)

[Figure 19: sequence diagram for generate report 69](#_Toc422269484)

[Finger 21: Deployment Diagram 74](#_Toc422269485)

[Finger 22: user interfaces login form 76](#_Toc422269486)

[Finger 23: user interfaces home pages 77](#_Toc422269487)

[Finger 24: user interfaces administrator pages 77](#_Toc422269488)

[Finger 25: user interfaces tourist pages 78](#_Toc422269489)

[Finger 26: user interfaces operator pages 78](#_Toc422269490)

## *****Abstract*****

*This project is a tour and travel guiding system for the Dessie tour guide agency which provides itinerary, hotel reservation and car rent, easily accessible information about destination along to the tourist. We have to data gathering method by using Interviewing, phone Interview and document analysis. Our project will be using object oriented system develop methodology. Total time required for the project completion is six months.*

## ****Keyword****

Keywords are words used as reference point for further information.

* Tourist: who travel to visit historical place.
* Administrator: who manage the system.
* Dessie tour guide Agency: client of the system.
* Operator: who use the application and give their services.
* User: someone to use the system (operator, tourist, admin).

## ****Abbreviation****

MySQL-My Structured Query Language

CSS - Cascading Style Sheet

WAMP- Windows, Apache, and MySQL

UML-Unified Modeling languages

SDLC-System Development Life Cycle

RAM-Read Access Memory

PHP-Personal Home Pages

CPU- Central Processing Unit

OS- operating system

ADMIN-administrator

LAN- Local Area Network

HTM- Hyper Text Language Mark-up Language

DGUI-Developing Graphical User Interface

BR-Business Rule

PDF-portable document format

MB/S-Mega Bits per Second

Uc-Use Case

CS4-Adope Photo Shop

Use case id-Use Case Identifies

GB-Gaga Bit

## CHAPTER ONE

## 1 System Requirement Specification

## 1.1 Introduction

The first section of this chapter presents details of system requirement specification. It includes Background of organization, statement of the project, significance of the system, scope and limitation system and methodology.

## *1.1.1 Background of the Organization*

Dessie tour guide agency is found in Samara city established in 2003 E.C. Dessie tour guide agency is providing information about the historical places for the tourist. In additional to direct or indirectly to give hotel reservation service and transport services.

Ethiopia has a rich wealth historic places such as Danakil depression, Hadar, Erta Ale, Awash National Park, Gondar palace, Axsum, Lalibela, Dillon Depression, sof omar cave, hammer and all cultures. Dessie tour guide agency provides general and specific information about historical place for the tourist.

## *1.1.2 Introduction about the Project*

Tour and travel guiding system is to develop and implement aweb based system for Dessie tour guide agency to handle tourist service for every historical place in Ethiopia

The project is intended help Dessie tour guide agency to simplify daily work, save time to provide information for tourist and decrease labor force.

The tour and travel guiding system will help the tourist to access the organization service online which saves cost and time.

## 1.2 Statement of the problem

Dessie tour guide agency use manual systems to process their business services, which create a great problem regarding with tourist and Dessie tour guide agency. This manual system consumes more time, cost and effort to accomplish task. The existing system of Dessie tour guide agency has following problems.

* The tourists spend more time to search information about the historical places, transportation and hotel management etc.
* The client system use stored data manually since there may be loss.
* The travel information is not easily access to tourist.
* Overhead cost because man power and phone
* A tourist does not get satisfaction for different kind of services such as hotel and transport services.
* It create poor guide experience to the tourist
* Delay work.
* High labor power to serve the tourist.

## 1.3 Significance of the system

This project brings some significance such as:-

* Tourist can get access to information in their needed
* Tourist can save their time to collect or access the necessary information. They don’t need to go ask to travel agent information, find operator, and find hotel and their service.
* Provides full and fast organized service for tourist like hotel reservation and provide view information about historical places.
* It reduces the number of man power needed.

## 1.4 Objective of the Project

## 1.4.1 **General objective**

The general objective of this project is to develop and implement web based system that will replace the manual system.

## 1.4**.2 Specific objective**

The project will have the following specific objectives.

* To provide Hotel reservation service for the tourist
* To provide service for the tourists in short time
* the tourist’s detail information register in the register form and create their profile .it is easy to search and apply for their service
* Tourist will be provide register online to get services
* To provide easily to Generate report for Dessie tour guide agency
* Dessie tour guide agency will be provide easily access tourist information.
* To improve the efficiency and to reduce the overload of work.
* To provide easy to search historical place and hotel service.

## 1.5 Scope and limitation of the project

## *1.5.1 Scope*

Scope of the project is describes as follows:-

* The system will support tourist to share their travel experience online
* Our system will improve the tour and travel guiding system by allowing like hotel reservation, provide information about historical places and itinerary.
* The system will be generating and record Tourist site to the tourist.
* System will be able to new tourist registration online details are kept within the database
* The system show difference packages based on price and quality
* System handles proper operator allocation to tourists.

## *1.5.2 Limitation*

Some of the limitations that will be placed on the development of the tour and travel guiding system include:-

* The system not provide guaranty for the tourist property
* The system doesn’t include online payment.
* It does not check tourist register it is true reservation or not.

## 1.6 Methodology

We have to develop our project use hardware and software requirement, data gathering and proposed and implementation method.

## *1.6.1 Data Gathering*

There is difference method for gathering data collection like interview, phones interview, Brainstorming, documentation, telephone and internet.

1. Interview: Through interview, we analyze the processes and procedures of the existing system and how try to understand the flow of tasks and services that are given to the tourist.

2. Brainstorming: Discuss all the details information based from the analysis. Discussion on combining knowledge/ideas to the Dessie tour guide agency for improving their web based system.

3. Documentation analyses is a process of using documented files for doing one task so we use document analysis to get more information about the tour guide system.

4. Phone interview: in addition to interview we contact the with organization and then exchange some ideas about their system how it has been working and the structure of this organization.

## *1.6.2 Design methodology*

For designing purpose we use object oriented system analysis and design. We have selected this because of the following advantages:-

* Objects are easier for people to understand,
* object are derived from the business that we are trying to automate
* Data and process are kept together the main reason we decide to use the object approach
* Code can be reused easily this means we are constantly looking for objects that would be useful in the system. Even if when a new system has minor differences s we are much more likely to be able to change our existing code to fit.
* Increased consistency among analysis, design and programming activities.
* Modeling techniques used in object oriented analysis are UML (Unified modeling language), that can present both static and dynamic behavior of the system

## *1.6.3 Analysis methodology*

This document we have to a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. From various types of SDLC methodologies, we choose the iterative methodology by considering the following advantages.

* Easy to plan and manage
* It is more flexible
* Intermediate product of each phase will be an input for the next phase.
* Risk management is possible

## 

## *1.6.4 Hardware and software used for Implementation methodology*

To implementation (developing) this project we used hardware and software tool.

*Hardware tools used for implementation*

* + 500 MB RAM
  + 400 GB hard disk
  + 16-Bit color display, 640x480 screen resolutions
  + Keyboard
  + Mouse
  + Personal Computer installed with windows 7 operating system
  + Removable Storage Medias, i.e. Flash Disk

***Software tools used for implementation***

* mysql server 6.01– to develop database application
* Wamp server - for web server software that can be run the PHP, CSS, and JavaScript language.
* Note pad++ or Dreamweaver 8 - to develop PHP ,HTML and JavaScript application
* Any browsers application

PHP and html tags for front end (developing Graphic user interface). PHP works great with html, interactive features and easy to learn. My SQL for back end.

## *1.6.5 Test methodology*

After the system is completed it will be tested using different testing method to test functionality. Those system testing methods are

* **Unit testing:** We will implement unit testing during development. So, if the error will occur it will fix immediately.
* **Integration testing:** We will make an integration testing to check whether the system meets all the functional requirements or not.
* **Alpha testing:** the system will tested by giving the correct input.
* **Beta testing:** In this testing method, the system is tested for incorrect data input.
* **development test**: system will be plan based on use case

## 

## CHAPTER TWO

## 2 Requirement Elicitations

## 2.1 The over view of the existing system

Dessie tour guide agency use manual systems to process their business services, which create a great problem regarding with tourist and Dessie tour guide agency. This manual system uses more time, cost and effort to accomplish the tasks. With the existing system the organization does not easily provide to the tourists such as hotel reservation, car reserve, and other facilities. From the existing system we have identifies the problem.

## *2.1.1 Problem of the existing system*

The problems that the organization faces in the existing system are the following:

* The tourists spend more time to search information about the historical places, transportation and hotel reservation etc.
* The organization system handle data manually since there may be loss of data and inconsistency among them.
* The travel information is not easily accessible to tourist.
* Overhead cost because man power and phone
* A tourist does not get satisfaction for different kind of services such as hotel and transport services.
* It create poor guiding experience to the tourist
* Delay work.
* High labor power to serve the tourist.

## *2.1.2 Weakness and strength of the existing system*

. Dessie tour guide agency use manual system to process their business services. The weakness that faces in the existing system is list below

* Delay work.
* High labor power to serve the tourist.
* It requires more amount of economy to facilitate.
* The tourists spend more time to search information about the tourist areas, transportation, facility, hotel reservation etc.
* The organization system handle data manually since there may be loss of data and inconsistency among them.

When we see strength of the Dessie tour guide agency are list below, but it is may not be support our system. Those are:-

* Use of internet access for the organization.
* The operator is knowledge based on the tour and exchanges their experience for other agency.
* There are many operator based on good experiences and ethical person.
* Good service for the tourist like car reservation and hotel reservation.

## *2.1.3 Business rules of the system*

Business rule are principles, requirements and polices that must be fulfilled and obligated in order the system will function properly and effectively. The business rules that must be considered for this project is described below.

**BR 1** The tourist must be register in the system to find any kind of services such as hotel reservation and transport can access by the tourist after the registration.

**BR 2** The tourist needs to visit the precious historical place they must be paid for that place.

**BR 3** The historic place can be visited in all kind of age that has not any restriction.

**BR.4** The tourist must be following the list of services, such as hotel and transport that the operator provides.

**BR.5** The tourist tends to choose the package offered by operator or ask directly to operator.

**BR.6** Tourist should be so polite and should be able to speak some global language natively.

**BR.7** Tourist must be respect rule and regulation for itinerary and their services to use properly.

## 2.2 overview of the proposed system

In proposed system we will describe the functional, non-functional and system requirement of the system. The proposed system is to develop and implement aweb based system for Dessie tour guide agency to handle tourist for every historical place in Amhara region.

## *2.2.1Functional requirement of the proposed system*

A functional requirement specifies particular results of a system. Functional requirement are expressed in the form of system must do requirement. There is some functional requirement that refer what module the system has: -

* The system will support tourist to share their travel experience online
* Our system will improve the tour and travel guiding system by allowing like hotel reservation, provide information about historical places and Tourist site.
* The system will be generating and record Tourist site to the tourist.
* Should be able to new tourist registration online details are kept within the database
* System has all the details of the authenticated tourist / operator
* Should be able to change the password and other details related to a system.
* System able to manages all the information of their all customers through databases
* The system show difference packages based on price and quality
* System handles proper operator allocation to tourists.
* System will car reserve
* The system can handle tourist travel schedule and service change
* System shall allow feedback suggestion and comment for tourist.
* Should be able to search reservation service
* System can manage user account easily

## *2.2.2 Nonfunctional requirement of the proposed system*

Nonfunctional requirement describe the user visible aspects of the system that are not directly related with the functional behavior the system. It can also support and provide more quality for the proposed system. Nonfunctional requirement in our proposed system deals with additional attributes such as:-

**Reliability:-**

* The system provides to the user correct information. When the user entered wrong inputs it notifies to correct the input data.
* The system shall require guide and help to be understood by user.
* The data and record that are saved shall be reused if needed.

**Portability:-**

* Ensure that the application or component can be easily and quickly ported to specified new environments if and when necessary.
* The system shall work properly in any browsers.

**Security:-**

* The system shall provide high level of security by blocking an authorised user to view secured system page.
* The system administrator must guarantee in revoke the privilege of user.
* The external security should be provided by giving login authentication.

**Performance:**

* The system shall minimize errors and should display clear error message that guides users.
* The system should be well fit to perform operations clearly without any problem. Client system accesses anything without error.
* The system have performed all functional key through single pressing button it response in short period of time.
* The system most host many users at same time with remote distance.
* The system should be retrieving data and it shall take a 2 minute.

**Usability:-**

* The end user will be able to access any page fast according to the internet connection speed is 3MB/S.
* The system will be available for 24 hours and 7 days a week.
* After the deployment of the project if any error occurs then it should be easily maintained by the web system developer.

**Flexibility:-**

* The operation shall be flexible and reports shall be presented in hardcopy, browser (html) format, and pdf.

## *2.2.3 System requirement of the proposed system*

**Development environment hardware:**

* + 500 MB RAM
  + 400 GB hard disk
  + 16-Bit color display, 640x480 screen resolutions
  + Keyboard
  + Mouse
  + Personal Computer installed with windows 7 operating system
  + Removable Storage Medias, i.e. Flash Disk
  + Network devices
  + Printer and copy machine
  + Un interrupted power supply

**Development environment software**:

* mysql server 6.01– to develop database application
* Wamp server - for web server software that can be run the php, css, and JavaScript language.
* Note pad++ or Dreamweaver 8 - to develop php ,html and JavaScript application
* Web browser- to search some information from the internet and for display registrar information.
* Microsoft word 2010 - To write document of project
* Microsoft power point 2010 – We use presentation purpose.
* E-draw: because it provide an easy way to draw UML diagrams
* Microsoft Visio 2013 and violet editor – To design Sequence Diagram , Class Diagram , Use case Diagram
* Adobe Photoshop CS4 – to read and write video and photo file
* U Lead - To edit photo and video

## 2.3 constraint and assumption

## *2.3.1 Constraints*

In the process of developing this project there were some constraints and challenges that limits to do what you went had been faced.

* Economic constraint:- to develop this project there may be finance shortage to purchase different materials and tools.
* Time constraint:- To complete this project there may be time limitation Better result will be achieved if the project period is lengthened
* Technical constraint:- To complete this project there may be different technical limitation
* Failure of electric power and internet connectivity- we try to solve this by taking back up to external storage devices.

## *2.3.2 Assumptions*

Risk assumptions are problems that occur when the team is doing the works consider like being true.

That includes:-

* Assume that the users know the basic computer skills.
* Assume that the hardware used to interact with the operating system will be a standard keyboard, mouse, monitor, and a standard printer for tacking hard copy of the reports generated as and when required.
* Assume that the users know how to start a program from the GUI, and can interact with a user interface using standard keyboard, mouse, and monitor.

## CHAPTER THREE

## 3 System Modeling

This chapter deals about the modeling techniques that follow to design and analysis the system. Object oriented analysis and design technique will be used. It includes system use case model, sequence diagram and class diagram.

## 3.1 Actor description

|  |  |  |
| --- | --- | --- |
| No. | Actors | Specification |
| 1 | Tourist | Who travel to visit historical place. Tourists are done difference activity to do in the system. |
| 2 | Operator | Who use the application and give their services. |
| 3 | Administrator(admin) | Who manages the organization an office and system |

# Table 1: Actor description

## 3.2 Use Case Identification

A representation of a system’s behavior or functionality under various conditions as the web based system responds to requests from Users. The team uses use case diagrams represents the major steps of the web based system function (process).

On the other hand use case diagrams are created to visualize the interaction of our system with external world and also a use case model is the representation of the system intended function and its environment. The functionality is specified by “actor” and “use case” specified the environment. Use case diagrams are made up of one or more users linked to one or more use cases. The actors are linked to their appropriate use cases through the line called association line. Actor refers to a persons of the system which uses built for achieve the goal.

## *3.2.1 Use case model*



Figure 1: use case diagram

## *3.2.2 Use case description*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-1 | | | |
| **Use Case Name:** | Add profile | | | |
| **Created By:** | Tour and travel guiding team | | **Last Updated By:** |  |
| **Date Created:** | 16/5/2009 E.C | | **Last Revision Date:** | 3/6/2009 E.C |
| **Actors:** | | **Primary Actor:** operator  **Supporting Actors: admin** | | |
| **Description:** | | Operator add profile into the system | | |
| **Preconditions:** | | Operator has member of the agency | | |
| **Normal Flow:** | | |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 | Operator open the system | The system  display the  home page form | | 2 | Operator click  add profile button | The system  display add profile  Form | | 3 | Operator fill the require information and click submit button | The system display save data success | | | |
| **Post conditions:** | | Operator is to guide the tourist and they have created user account. | | |
| **Alternative Flows:** | | 3a. In line 3 of the normal flow, if the operator is not fill correct information   |  |  |  | | --- | --- | --- | | Line | System actor | System response | | 1 |  | System display a  message invalid fill  information and  please enter another  information |   restart at line 3 of the normal flow | | |
| **Includes:** | | No includes | | |
| **Frequency of Use:** | | Add profile once a year | | |
| **Special Requirements:** | | Reliability, usability, Performance requirements | | |
| **Assumptions:** | | For add profile Use Case, an assumption could be: system only understands English language. Operator is member of the organization. | | |
| **Notes and Issues:** | | Maximum size of fill data is 40 bytes | | |

# Table 2: use case description for add profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-2 | | | |
| **Use Case Name:** | Register | | | |
| **Created By:** | Tour and travel guiding team | | **Last Updated By:** |  |
| **Date Created:** | 16/5/2009 E.C | | **Last Revision Date:** | 3/6/2009 E.C |
| **Actors:** | | **Primary Actor:** tourist  **Supporting Actors:** no | | |
| **Description:** | | Tourist must be register into the system | | |
| **Preconditions:** | | Tourist open the system home page form | | |
| **Normal Flow:** | | |  |  |  | | --- | --- | --- | | Line | System actor action | System responds | | 1 | Tourist open the system | The System display home page Form | | 2 | Tourist click register button | The System display register Form | | 3 | Tourist insert the information in register form | The System recorded insert information | | 4 | Tourist click submit button | Systems save data into database and display successfully and system send user account. | | | |
| **Post conditions:** | | The system display registers successfully for the tourist and create account. | | |
| **Alternative Flows:** | | 3a. If the line 3 of the normal flow, if the tourist is insert  Invalid information.   |  |  |  | | --- | --- | --- | | Line | System actor | System response | | 1 |  | The system display  a message invalid insert data And fill another data |   restart at line 3 of the normal flow | | |
| **Includes:** | | No includes | | |
| **Frequency of Use:** | | Tourists registered 20 per day, 220 per week and 2000 per year. | | |
| **Special Requirements:** | | Reliability, usability, Performance requirements | | |
| **Assumptions:** | | For register Use Case, an assumption could be:  system only understands English language. Tourist must be visit by this agency | | |
| **Notes and Issues:** | | Maximum size of insert data is 40 bites | | |

# Table 3: use case description for registration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-3 | | | |
| **Use Case Name:** | Login | | | |
| **Created By:** | Tour and travel guiding team | | **Last Updated By:** |  |
| **Date Created:** | 16/5/2009 E.C | | **Last Revision Date:** | 3/6/2009 E.C |
| **Actors:** | | **Primary Actor:** tourist, administrator, operator (user)  **Supporting Actors:** administrator, operator, tourist (user) | | |
| **Description:** | | It allow the existing user to login | | |
| **Preconditions:** | | User must have valid user name, user type and password | | |
| **Normal Flow:** | | |  |  |  | | --- | --- | --- | | Line | Actor action | System response | | 1 | user open the system and click login form | System display home  form and display login  form | | 2 | user insert his user name, user type and password | The system record user name, user type and password | | 3 | User view the available form | The system display the request available information | | | |
| **Post conditions:** | | The system will be login, and then open other form. | | |
| **Alternative Flows:** | | 2a. In line 2 of the normal flow, if the user fill invalid user name, password and user types   1. Message to user to re-enter user name, password and user type 2. Enters fill correct user name, password and user types 3. Use Case resumes on line 3 of normal flow | | |
| **Includes:** | | Login | | |
| **Frequency of Use:** | | Login into the system 10 per hour and 220 per day. | | |
| **Special Requirements:** | | Performance, reliability, portability, usability, and Security requirements | | |
| **Assumptions:** | | For login Use Case, an assumption could be: system understands English language. All users have their own account. | | |
| **Notes and Issues:** | | Maximum size of fill data is 40 bytes | | |

# Table 4: use case description for login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-4 | | | |
| **Use Case Name:** | Add car | | | |
| **Created By:** | Tour and travel guiding team | | **Last Updated By:** |  |
| **Date Created:** | 16/5/2009 E.C | | **Last Revision Date:** | 3/6/2009 E.C |
| **Actors:** | | **Primary Actor:** administrator  **Supporting Actors:** | | |
| **Description** | | administrator allow to add the car | | |
| **Preconditions:** | | Administrator must be login the system | | |
| **Normal Flow:** | | |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 | Administrator open the System | System display the  homepage form | | 2 | Administrator click login button to the system, then login | System display login form,  Then systems display another form. | | 3 | Administrator click add car button | System display the add car form | | 4 | Administrator fill the require data into add car form, then submit | The system display the  add car data and save into  the database | | | |
| **Post conditions:** | | New car add in to the system | | |
| **Alternative Flows:** | | 2a. In line 2 of the normal flow, if the administrator fill invalid username, user type and password   1. Message to user to re-enter login form 2. Enter correct user name, password and user types 3. Use Case resumes on line 3 of normal flow   4a. In line 4 of the normal flow, if the Administrator is not fill correct data about car   |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 |  | System display a message add car is not valid, please insert other data |   restart at line 4 of the normal flow | | |
| **Includes:** | | Login | | |
| **Frequency of Use:** | | Add car per year and once month | | |
| **Special Requirements:** | | Performance, reliability, portability, usability, and Security requirements | | |
| **Assumptions:** | | For add car Use Case, an assumption could be: system understands English language. Car will be added in the system. They have organization. | | |
| **Notes and Issues:** | | Maximum size of fill data is 40 bytes | | |

# Table 5: use case description for add car

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-5 | | | |
| **Use Case Name:** | update car | | | |
| **Created By:** | Tour and travel guiding team | | **Last Updated By:** |  |
| **Date Created:** | 16/5/2009 E.C | | **Last Revision Date:** | 3/6/2009 E.C |
| **Actors:** | | **Primary Actor:** administrator  **Supporting Actors:** | | |
| **Description:** | | administrator allow to update the car | | |
| **Preconditions:** | | Administrator must be login the system | | |
| **Normal Flow:** | | |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 | Administrator open the System | System display the  homepage form | | 2 | Administrator click login button to the system, then login | System display login form,  Then system displays another form. | | 3 | Administrator click update car button | System display the update  car form | | 4 | Administrator fill the require data into update car form then submit | The system display the  update car data and save  into the database | | | |
| **Post conditions:** | | New car update in to the system | | |
| **Alternative Flows:** | | 2a. in line 2 of the normal flow, if the administrator fill invalid username, user type and password   1. Message to user to re-enter login form 2. Enters correct user name, password and user types 3. Use Case resumes on line 3 of normal flow   4a. In line 4 of the normal flow, if the admin is not in the fill correct car update information   |  |  |  | | --- | --- | --- | | Line | System actor | System response | | 1 |  | The system display a error message update car is incorrect .please insert another data |   restart at line 4 of the normal flow | | |
| **Includes:** | | Login | | |
| **Frequency of Use:** | | update car per year and once month | | |
| **Special Requirements:** | | Reliability, usability, Portability, Security and performance requirement | | |
| **Assumptions:** | | For update car Use Case, an assumption could be:  system understands English language. If the admin update car first they will be add car. The car must have the organization. | | |
| **Notes and Issues:** | | Maximum size of fill data is 40 bytes | | |

# Table 6: use case description for update car

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-6 | | | |
| **Use Case Name:** | delete car | | | |
| **Created By:** | Tour and travel guiding team | | **Last Updated By:** |  |
| **Date Created:** | 16/5/2009 E.C | | **Last Revision Date:** | 3/6/2009 E.C |
| **Actors:** | | **Primary Actor:** administrator  **Supporting Actors:** | | |
| **Description:** | | administrator allow to delete the car | | |
| **Preconditions:** | | Administrator must be login the system | | |
| **Normal Flow:** | | |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 | Administrator open the System | System display the  homepage form | | 2 | Administrator click login button to the system, then login | System display login form,  then system display  another form. | | 3 | Administrator click delete car button | System display the delete  car form | | 4 | Administrator fill the require data into delete car form, then delete button | The system display the Delete car data and delete into the database | | | |
| **Post conditions:** | | car delete form the system | | |
| **Alternative Flows:** | | 2a. line 2 of the normal flow, if the administrator fill invalid username, user type and password   1. Message to user to re-enter login form 2. Enters fill correct user name, password and user types 3. Use Case resumes on line 3 of normal flow   4a. In line 4 of the normal flow, if the admin is not insert valid data about car   |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 |  | The system display message “are you sure Car is delete” into the database and asking the administrator |   restart at line 4 of the normal flow | | |
| **Includes:** | | Login | | |
| **Frequency of Use:** | | Delete car per year and once month | | |
| **Special Requirements:** | | Security, Reliability, usability, portability, Performance requirements | | |
| **Assumptions:** | | For delete car Use Case, an assumption could be:  system understands English language. If admin delete the car, first car must be added into the system. the car have the organization | | |
| **Notes and Issues:** | | Maximum size of fill data is 40 bytes | | |

# Table 7: use case description for delete car

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC 7 | | | |
| **Use Case Name:** | Manage tourist site | | | |
| **Created By:** | Tour and travel guiding team | | **Last Updated By:** |  |
| **Date Created:** | 16/5/2009 E.C | | **Last Revision Date:** | 3/6/2009 E.C |
| **Actors:** | | **Primary Actor:** administrator  **Supporting Actors:** | | |
| **Description:** | | administrator allow to manage tourist site | | |
| **Preconditions:** | | administrator must be login the system | | |
| **Normal Flow:** | | |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 | Administrator open the System | System display the  homepage form | | 2 | Administrator click login button to the system, then login | System display login form,  then system display  another form. | | 3 | Administrator click add historical place button | System display the add  historical place form | | 4 | Administrator fill the require data into manage tourist site form, then submit the required data | The system display the  Manage tourist sitedata  and save into the database | | | |
| **Post conditions:** | | New tourist site add and update in to the system | | |
| **Alternative Flows:** | | 2a. In line 2 of the normal flow, if the administrator fill invalid username, user type and password   1. Message to user to re-enter login form   2 Enters fill correct user name, password and user types  3 Use Case resumes on line 3 of normal flow  4a. In line 4 of the normal flow, if the administrator is not fill correct  Data about tourist site   |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 |  | System display a message add and update tourist site is not valid, please insert other data |   restart at line 4 of the normal flow | | |
| **Includes:** | | Login | | |
| **Extended’s:** | | Update schedule and view itinerary | | |
| **Frequency of Use:** | | manage tourist site per year and once month | | |
| **Special Requirements:** | | Security, Reliability, usability, portability, Performance requirements | | |
| **Assumptions:** | | For manage tourist site Use Case, an assumption could be:  system understands English language. Manage tourist site in the data base if site are most attractive for tourist. | | |
| **Notes and Issues:** | | Maximum size of fill data is 255 bytes | | |

# Table 8: use case description for manage tourist site

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-8 | | | |
| **Use Case Name:** | Assign operator | | | |
| **Created By:** | Tour and travel guiding team | | **Last Updated By:** |  |
| **Date Created:** | 16/5/2009 E.C | | **Last Revision Date:** | 3/6/2009 E.C |
| **Actors:** | | **Primary Actor:** administrator  **Supporting Actors:** tourist | | |
| **Description:** | | Administrator assign the operator based on the tourist needed | | |
| **Preconditions:** | | The information must exist in the system | | |
| **Normal Flow:** | | |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 | Administrator open the system | System will be display home page form | | 2 | Administrator click admin login button, then login | System display login form,  Then system displays another form. | | 3 | Administrator click the assign operator button | System display the  assign operator form | | 4 | Administrator fill assign the operator based on the tourist needed | System display assign  Operator name and  display the message  success compilation | | | |
| **Post conditions:** | | operator will be assigned to the tourist needed to visit a place | | |
| **Alternative Flows:** | | 2a. In line 2 of the normal flow, if the administrator fill invalid username, password and user types   1. Message to tourist to re-enter login form 2. Enters fill correct user name, password and user types 3. Use Case resumes on line 3 of normal flow   4a. In line 4 of the normal flow, if the administrator is not valid fill assign operator   |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 |  | System display message will be invalid assign operator, please assign  anther operator |   restart at line 4 of the normal flow | | |
| **Includes:** | | Login | | |
| **Frequency of Use:** | | Assign operator 5 per hour, 10 per day, once a week. | | |
| **Special Requirements:** | | Reliability, usability, Performance, portability and security requirement | | |
| **Assumptions:** | | For assign operator Use Case an assumption could be: system understands English language. Administrator is assign Operator for the visiting program based on the tourist needed if the operator is a member of organization (add profile). | | |
| **Notes and Issues:** | | Maximum size of fill data is 40 bytes | | |

# Table 9: use case description for assign operator

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-9 | | | |
| **Use Case Name:** | Car reservation | | | |
| **Created By:** | Tour and travel guiding team | | **Last Updated By:** |  |
| **Date Created:** | 16/5/2009 E.C | | **Last Revision Date:** | 3/6/2009 E.C |
| **Actors:** | | **Primary Actor:** tourist  **Supporting Actors:** administrator | | |
| **Description:** | | Tourist must be allow register | | |
| **Preconditions:** | | tourist must be login the system | | |
| **Normal Flow:** | | |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 | tourist open the  System | System display the  homepage form | | 2 | Tourist click login button, then login | The system display login form, then display another form | | 3 | Tourist click car reservation button | System display the car reservation form | | 4 | tourist fill the require data into car reservation form | The system display the Car reservation data and  submit into the database | | | |
| **Post conditions:** | | Tourist will be car reservation | | |
| **Alternative Flows:** | | 2a. In line 2 of the normal flow, if the tourist fill invalid username, user type and password   1. Message to user to re-enter login form 2. Enters fill correct user name, password and user types 3. Use Case resumes on line 3 of normal flow   4a. In line 4 of the normal flow, if the tourist is fill invalid data for the reservation form   |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 |  | System display a message fill require data is invalid, please insert another car reserve |   restart at line 4 of the normal flow | | |
| **Includes:** | | Login | | |
| **Frequency of Use:** | | Car reservation 5 per hour, 10 per day, once a week. | | |
| **Special Requirements:** | | Security, Reliability, usability, portability and Performance requirements | | |
| **Assumptions:** | | System only understands English language. The car reserved from the system. Car reserve in per day. Car must be added into the system. | | |
| **Notes and Issues:** | | Maximum size of fill data is 20 bytes | | |

# Table 10: use case description for car reservation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-10 | | | |
| **Use Case Name:** | Hotel reservation | | | |
| **Created By:** | Tour and travel guiding team | | **Last Updated By:** |  |
| **Date Created:** | 16/5/2009 E.C | | **Last Revision Date:** | 3/6/2009 E.C |
| **Actors:** | | **Primary Actor:** tourist  **Supporting Actors:** | | |
| **Description:** | | Searching tourist information for the hotel. | | |
| **Preconditions:** | | The tourists need to get information about hotel. | | |
| **Normal Flow:** | | |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 | tourist open the System | System display the  homepage form | | 2 | Tourist click login button, then login | The system display login form, then display another form | | 3 | Tourist click hotel reservation button | The system display hotel  List form | | 4 | Tourist select the hotel from the hotel reservation form | The system display select hotel | | 5 | tourist fill the require services to the hotel form | The system check the require service form the hotel. | | 6 | Tourist search the available information in the external database | System display search data | | | |
| **Post conditions:** | | The system displays the required information successfully. | | |
| **Alternative Flows:** | | 2a. In line 2 of the normal flow, if the tourist fill invalid username, user type and password   1. Message to user to re-enter login form 2. Enters fill correct user name, password and user types 3. Use Case resumes on line 3 of normal flow   5a. In line 5 of the normal flow, if the tourist is not fill the available information   |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 |  | The system display message service is not available, please search other hotel |   restart at line 5 of the normal flow | | |
| **Includes:** | | Login | | |
| **Frequency of Use:** | | Hotel reservation 20 per hour, 220 per day, once a week. | | |
| **Special Requirements:** | | Security, Reliability, usability, portability and Performance requirements | | |
| **Assumptions:** | | For hotel reservation Use Case, an assumption could be: the system only understands English language. There is external database that can be add, update and delete hotel services(does not matter our system in the hotel).hotel must be link  Into the system. From the hotel only room reserve. | | |
| **Notes and Issues:** | | Maximum size of fill data is 40 bytes | | |

# Table 11: use case description for hotel reservation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-11 | | | |
| **Use Case Name:** | Tourist site reservation | | | |
| **Created By:** | Tour and travel guiding team | | **Last Updated By:** |  |
| **Date Created:** | 16/5/2009 E.C | | **Last Revision Date:** | 3/6/2009 E.C |
| **Actors:** | | **Primary Actor:** tourist  **Supporting Actors:** admin | | |
| **Description:** | | Searching tourist information for the tourist site. | | |
| **Preconditions:** | | The tourists need to get information about tourist site. | | |
| **Normal Flow:** | | |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 | tourist open the System | System display the  homepage form | | 2 | Tourist click login button, then login | The system display login form, then display another form | | 3 | Tourist click tourist site reservation button | The system display tourist site list form | | 4 | Tourist select the tourist site from the tourist site reservation form | The system display select tourist site | | 5 | tourist fill the require the date in the tourist site reservation form | The system check the  require information. | | 6 | Tourist search the available information from database | System display search data | | | |
| **Post conditions:** | | The system displays the required information successfully. | | |
| **Alternative Flows:** | | 2a. In line 2 of the normal flow, if the tourist fill invalid username, user type and password   1. Message to user to re-enter login form 2. Enters fill correct user name, password and user types 3. Use Case resumes on line 3 of normal flow   5a. In line 5 of the normal flow, if the tourist is not fill the available information   |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 |  | The system display message service is not available, please search other tourist site reservation |   restart at line 5 of the normal flow | | |
| **Includes:** | | Login | | |
| **Frequency of Use:** | | Tourist site reservation 20 per hour, 220 per day, once a week. | | |
| **Special Requirements:** | | Security, Reliability, usability, portability and Performance requirements | | |
| **Assumptions:** | | For tourist site reservation Use Case, an assumption could be: the system only understands English language. First manage the tourist site into system. | | |
| **Notes and Issues:** | | Maximum size of fill data is 40 bytes | | |

# Table 12: use case description for tourist site reservation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-12 | | | |
| **Use Case Name:** | Update schedule | | | |
| **Created By:** | Tour and travel guiding team | | **Last Updated By:** |  |
| **Date Created:** | 16/5/2009 E.C | | **Last Revision Date:** | 3/6/2009 E.C |
| **Actors:** | | **Primary Actor:** tourist  **Supporting Actors:** | | |
| **Description:** | | tourist allow to update the tourist visiting program | | |
| **Preconditions:** | | tourist must be login the system and extend in tourist site reservation | | |
| **Normal Flow:** | | |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 | tourist open the System | System display the  homepage form | | 2 | Tourist click login button, then login | The system display login form, then display another form | | 3 | Tourist click tourist site reservation button | The system display tourist site list form | | 4 | Tourist select the tourist site from the tourist site reservation form | The system display select tourist site | | 5 | tourist fill the require the date in the tourist site reservation form | The system check the  require information. | | 6 | Tourist search the available information from database | System display search data | | 7 | Tourist click update schedule form button and fill the update schedule form from extend to the tourist site reservation form | System display update schedule form, then system  display schedule is success. | | | |
| **Post conditions:** | | Fare and time of journey will be update schedule | | |
| **Alternative Flows:** | | 2a. In line 2 of the normal flow, if the tourist fill invalid username, user type and password   1. Message to user to re-enter login form 2. Enters fill correct user name, password and user types 3. Use Case resumes on line 3 of normal flow   5a. In line 5 of the normal flow, if the tourist is not fill the available information   |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 |  | The system display message service is not available, please search other tourist site |   restart at line 5 of the normal flow  7a. In line 7 of the normal flow, if the tourist is not input correct information about update schedule form   |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 |  | The system display error  message and enter other  data in the update schedule form |   restart at line 7 of the normal flow | | |
| **Includes:** | | Login and tourist site reservation | | |
| **Frequency of Use:** | | Update schedule 5 per hour, 10 per day, once a week. | | |
| **Special Requirements:** | | Security, Reliability, usability, portability and Performance requirements | | |
| **Assumptions:** | | For update schedule Use Case, an assumption could be:  system understands English. Tourist update schedule based on the needed. | | |
| **Notes and Issues:** | | Maximum size of fill data is 255 bytes | | |

# Table 13: use case description for update schedules

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-13 | | | |
| **Use Case Name:** | View hotel reservation | | | |
| **Created By:** | Tour and travel guiding team | | **Last Updated By:** |  |
| **Date Created:** | 16/5/2009 E.C | | **Last Revision Date:** | 3/6/2009 E.C |
| **Actors:** | | **Primary Actor:** Operator  **Supporting Actors:** tourist | | |
| **Description:** | | Operator view the tourist reservation hotel | | |
| **Preconditions:** | | The operator view the information about the hotel where the tourist they reserve and register into the system. | | |
| **Normal Flow:** | | |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 | Operator open the system | The system display the  home page form | | 2 | Operator click login button, then login | The system display login form, then display another form | | 3 | Operator click the view hotel reservation form | The system display the  view of information to  reserve tourist hotel | | 4 | Operator click search button and fill require data in view hotel reservation | The system display  the search information | | | |
| **Post conditions:** | | The system displays the required information successfully. | | |
| **Alternative Flows:** | | 2a. In line 2 of the normal flow, if the operator fill invalid username, user type and password   1. Message to user to re-enter login form 2. Enters fill correct user name, password and user types 3. Use Case resumes on line 3 of normal flow   4a. In line 4 of the normal flow, if the tourist is not fill the required information   |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 |  | The system display a message indicating that search data is not available And search another tourist  to reserve hotel |   Restart line 4 in the normal flow | | |
| **Includes:** | | Login | | |
| **Frequency of Use:** | | View hotel reservation 20 per hour,100 per day and  once week | | |
| **Special Requirements:** | | Portability, reliability, usability, Security and Performance requirement. | | |
| **Assumptions:** | | For hotel reservation Use Case, an assumption could be: the system only understands English language. There is external database that can be add, update and delete hotel services (does not matter our system in the hotel).hotel must be link Into the system and view reserve hotel in the tourist. First tourist must be reserve. | | |
| **Notes and Issues:** | | Maximum size of fill data is 40 bytes | | |

# Table 14: use case description view hotel reservation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-14 | | | |
| **Use Case Name:** | View tourist site reservation | | | |
| **Created By:** | Tour and travel guiding team | | **Last Updated By:** |  |
| **Date Created:** | 16/5/2009 E.C | | **Last Revision Date:** | 3/6/2009 E.C |
| **Actors:** | | **Primary Actor:** operator  **Supporting Actors:** tourist | | |
| **Description:** | | operator view the in the home page | | |
| **Preconditions:** | | The operator views the information about the tourist site. | | |
| **Normal Flow:** | | |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 | operator open the system | The system display the  home page form | | 2 | Operator click login button, then login | The system display login form, then display another form | | 3 | operator click the view tourist site reservation form | The system display the  view information to tourist site reservation | | 4 | operator click search button and fill require data to the tourist and identifies the tourist site | The system display  the search information | | | |
| **Post conditions:** | | The system displays the required information successfully. | | |
| **Alternative Flows:** | | 2a. In line 2 of the normal flow, if the operator fill invalid username, user type and password   1. Message to user to re-enter login form 2. Enters fill correct user name, password and user types 3. Use Case resumes on line 3 of normal flow   4a. In line 4 of the normal flow, if the operator is not fill the required information   |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 |  | The system display a message indicating that search data is not available  and search another view tourist site reserve |   Restart line 4 in the system | | |
| **Includes:** | | Login | | |
| **Frequency of Use:** | | View tourist site reservation 60 per hour and 300 per day | | |
| **Special Requirements:** | | Reliability, security, portability, usability and Performance requirement. | | |
| **Assumptions:** | | For view reservation Use Case, an assumption could be:  system understands English language. Operator views the tourist site in the system database. First tourist site reservation by the tourist. | | |
| **Notes and Issues:** | | Maximum size of fill data is 40 bytes | | |

# Table 15: use case description for view tourist site reservation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-15 | | | |
| **Use Case Name:** | Comment | | | |
| **Created By:** | Tour and travel guiding team | | **Last Updated By:** |  |
| **Date Created:** | 16/5/2009 E.C | | **Last Revision Date:** | 3/6/2009 E.C |
| **Actors:** | | **Primary Actor:** tourist  **Supporting Actors:** administrator | | |
| **Description:** | | Tourist comment into system | | |
| **Preconditions:** | | Tourist must have their account | | |
| **Normal Flow:** | | |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 | Tourist open the system | The System will be display  the home page | | 2 | Tourist click login button, Then login | The system display login form, then display another form | | 3 | Tourist clicks the comment button. | The System display the comment form | | 4 | Tourist fill the comment if they submit | The System recorded data and display the success comment | | | |
| **Post conditions:** | | Tourist’s comment is successfully | | |
| **Alternative Flows:** | | 2a. In line 2 of the normal flow, if the tourist fill invalid username, user type and password   1. Message to user to re-enter login form 2. Enters fill correct user name, password and user types 3. Use Case resumes on line 3 of normal flow   4a. In line 4 of the normal flow, if the tourist is fill invalid comment   |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 |  | The system display the a message comment file is error, please enter other data. |   Restart line 4 in the normal flow | | |
| **Includes:** | | Login | | |
| **Frequency of Use:** | | Tourist comment 20 per hour and 220 per day. | | |
| **Special Requirements:** | | Security, Reliability, usability, portability and Performance requirements | | |
| **Assumptions:** | | For comment Use Case, an assumption could be:  system only understands English language. | | |
| **Notes and Issues:** | | Maximum size of fill data is 255 bytes | | |

# Table 16: use case description for comment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-16 | | | |
| **Use Case Name:** | Create operator account | | | |
| **Created By:** | Tour and travel guiding team | | **Last Updated By:** |  |
| **Date Created:** | 16/5/2009 E.C | | **Last Revision Date:** | 3/6/2009 E.C |
| **Actors:** | | **Primary Actor:**  administrator  **Supporting Actors:** operator | | |
| **Description:** | | Create account for different operator and tourist. | | |
| **Preconditions:** | | Admin must be to login. | | |
| **Normal Flow:** | | |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 | Administrator open the System | System display the  homepage form | | 2 | Administrator click login button, then login | The system display login form, then display another form | | 3 | Administrator click create account button | System display the Create operator account form | | 4 | Administrator fill the require data into create account form | The system display the  Create account data | | 5 | Administrator click save button | The display create account  is success and save Database | | | |
| **Post conditions:** | | Account creation performs successfully. | | |
| **Alternative Flows:** | | 2a. In line 2 of the normal flow, if the admin fill invalid username, user type and password   1. Message to user to re-enter login form 2. Enters fill correct user name, password and user types 3. Use Case resumes on line 3 of normal flow   4a. In line 4 of the normal flow, if the admin is not fill the required information.   |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 |  | The system display a message account is invalid and please create another account |   Restart line 4 in the normal flow | | |
| **Includes:** | | Login | | |
| **Frequency of Use:** | | Admin create operator account 20 per hour, 220 per day and once week | | |
| **Special Requirements:** | | Security, Reliability, usability, portability and Performance requirements | | |
| **Assumptions:** | | For create operator account Use Case, an assumption could be:  the system only understands English language. Admin create difference for the user and user must be register and add profile. | | |
| **Notes and Issues:** | | Maximum size of fill data is 20 bytes | | |

# Table 17: use case description for account create

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-17 | | | |
| **Use Case Name:** | Delete operator account | | | |
| **Created By:** | Tour and travel guiding team | | **Last Updated By:** |  |
| **Date Created:** | 16/5/2009 E.C | | **Last Revision Date:** | 3/6/2009 E.C |
| **Actors:** | | **Primary Actor:**  administrator  **Supporting Actors:** operator | | |
| **Description:** | | Delete account for different operator. | | |
| **Preconditions:** | | Admin login in to the system | | |
| **Normal Flow:** | | |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 | Administrator open the System | The system display  the homepage form | | 2 | Administrator click login button, then login | The system display login form, then display another form | | 3 | Administrator click delete operator account button | The system display the  Delete operator account form | | 4 | Administrator fill the require data into delete operator account form | The system display the  delete account data | | 5 | Administrator click delete button | The system display delete operator account is success and delete data  from database | | | |
| **Post conditions:** | | Delete operator account perform successfully. | | |
| **Alternative Flows:** | | 2a. In line 2 of the normal flow, if the admin fill invalid username, user type and password   1. Message to user to re-enter login form 2. Enters fill correct user name, password and user types 3. Use Case resumes on line 3 of normal flow   4a. In line 4 of the normal flow, if the admin is not fill the required information.   |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 |  | The system display a message delete account is invalid and please delete another account ”are you sure delete the account” |   Restart line 4 in the normal flow | | |
| **Includes:** | | Login | | |
| **Frequency of Use:** | | Admin delete operator account 20 per hour, 220 per day and once week | | |
| **Special Requirements:** | | Security, Reliability, usability, portability and Performance requirements | | |
| **Assumptions:** | | For delete operator account Use Case, an assumption could be:  system understands English language. Admin delete account if user is out of the system. First account must be creating. | | |
| **Notes and Issues:** | | Maximum size of fill data is 255 bytes | | |

# Table 18: use case description for delete account

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-18 | | | |
| **Use Case Name:** | Generate report | | | |
| **Created By:** | Tour and travel guiding team | | **Last Updated By:** |  |
| **Date Created:** | 16/5/2009 E.C | | **Last Revision**  **Date:** | 3/6/2009 E.C |
| **Actors:** | | **Primary Actor:** administrator  **Supporting Actors:** operator | | |
| **Description:** | | If the information of some activity or some record or detail is needed, the system generates report. | | |
| **Preconditions:** | | Administrator must be login the system | | |
| **Normal Flow:** | | |  |  |  | | --- | --- | --- | | Line | System actor action | System response | | 1 | Administrator open the system | System will be  display homepage form | | 2 | Administrator click login button, then login | The system display login form, then display another form | | 3 | Administrator click generate report button | System display the report | | 4 | Administrator click print | The system print the report | | | |
| **Post conditions:** | | The system generated the report | | |
| **Alternative Flows:** | | 2a. In line 2 of the normal flow, if the administrator fill invalid username, user type and password   1. Message to user to re-enter login form 2. Enters fill correct user name, password and user types 3. Use Case resumes on line 3 of normal flow | | |
| **Includes:** | | Login | | |
| **Frequency of Use:** | | Generate report once a week, once a month and once a year | | |
| **Special Requirements:** | | Flexibility, security, Reliability, usability, portability and Performance requirements | | |
| **Assumptions:** | | For generate report Use Case, an assumption could be:  system only understand English language. Report display in English language. The report generated amount of income and how much tourist visit in tourist site per month and year. | | |
| **Notes and Issues:** | | Maximum size of fill data is 255 bytes | | |

# Table 19: use case description for generate report

## 3.3 sequence diagram

Sequence diagrams are used to formalize the behavior of this system and to visualize the communication among objects. The diagrams show the flow of messages from one object to another, and as such correspond to the methods and events supported by a class/object. These typically show a user or actor, and the objects and components they interact with in the execution of a use case. The following describes the sequence diagrams of the tour and travel guiding. A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.



### Figure 2: sequence diagram for add profile



Figure 3: sequence diagram for registration



### Figure 4: sequence diagram to login



Figure 5: sequence diagram for add car



Figure 6: sequence diagram for update car



Figure 7: sequence diagram for delete car



Figure 8: sequence diagram for manage tourist site

Figure 9: sequence diagram for assign operator



Figure 10: sequence diagram for car reservation



Figure 11: sequence diagram for hotel reservation



Figure 12: sequence diagram to tourist site reservation



Finger 13: sequence diagram for update schedule



Figure 14: sequence diagram for view hotel reservation



Figure 15: sequence diagram for view tourist site reservation



Figure 16: sequence diagram for comment



Finger 17: sequence diagram for create operator account



Finger 18: sequence diagram for delete operator account



Figure 19: sequence diagram for generate report

## 3.4 Class diagram

In class modeling the static structure of the web page will be built. In particular, it shows the things such as classes and types, their internal structure and relationships, to other classes.Finger 20: class diagram

## 

## CHAPTER FOUR

## 4 System Design

System designing deal with how the system is to be looks like after the analysis is successfully completed: mean that functionality, non-functionality and analysis modeling is clearly achieved the next step is followed by system designing. It shows also the computerized part of the system. System design helps the programmer to configure the implementation or coding with the customer or user of the software.

## 4.1 Design Goal

Design goal describes the quality of the system that developers should be optimize such goal are normally derived from non-functional requirement of the system. It’s grouped in to four categories. There are:-

1. Performance
2. Dependability
3. Maintenance
4. End user criteria
5. Performance

The part of the system to be used for the tourist should have a fast response time (real time) with maximum throughput. Furthermore, the system should not be taking up too much space in memory. The users of the system have chosen fast response time over throughput and hence the system should try to be more interactive.

1. Dependability

The system should be highly dependable as it’s expected to be used by non IT professionals. The system should be robust and fault tolerant.

1. Maintenance

The system should be extensible to add new functionalities at later stage. It should also be easily modifiable to make changes to the features and functionalities.

1. End user criteria

Usability:- is the extent to which a product can be used by any user to achieve specified goal with effectiveness, efficiency and satisfaction in specified context of use from the end user perspective the system should be designed in such a way that it’s easy to learn and use efficient and having few errors if any Trade-off is in evitable in trying to achieve a particular design goal.

## 4.2 System Decomposition and Interrelation between them

System decomposition deals with identification of subsystems, services and their relationship to each other and to the environment. In object-oriented design, this consists of Defining the software architecture and Mapping analysis objects into the architecture’s subsystems. This decomposition shows correspondence between requirements and elements of the constructed system and addresses emergent, non-functional requirements by satisfying design goals. Subsystem decompositions will help us to reduce the complexity of the system. So the team identifies the following subsystem from the main system:

* Tourist site management
* Operator account management
* Car
* Operator
* Tourist site reservation
* Hotel
* Tourist register

The Operator account management subsystems control the account of the system Operator account and tourist site management control and manage tourist site. The other subsystem is reservations which provides and access information about tourist and reserve their site. The car and hotel use tourist reserve hotel and car in the system.

**4.3 System Architecture**

Current system of Dessie tour guide agency system use manual system so, there is no have any software architecture. ”the central repository of the proposed system is mysql database server which every data related with the system.”

The term system architecture is used to describe the overall design and structure of a computer network or system. It includes a wide range of physical devices, a method is required to organize and connect these items together in a cohesive manner. The term is also used to describe complex computer\_ software tools. Systems Architecture is a generic discipline to handle objects (existing or to be created) called "systems", in a way that supports reasoning about the structural properties of these objects. A **system architecture** or **systems architecture** is the conceptual model that defines the structure, behavior, and more views of a system. There are three main components to any system architecture of the system theses includes: storage, connectivity, and user experience. It is important to note that system architecture must be flexible and able to meet changing needs quickly. A structure should not be very rigid and it will not be able to support or facilitate new software or hardware.

Processing power is based on the computer or server. This hardware is akin to the brain of the system. Purchasing and installing the correct allocation of processors to the system must be based on the software specifications, number of concurrent users, strength of the connection, and applications. When designing a system, scalability is critical. The system architecture must allow additional processors to be added without any interruption to the current structure.

## 4.4 Deployment Diagram

A UML deployment diagram depict a view of run-time configuration of processing nodes and component that run on those nodes, in other words deployment diagram on those nodes show the hardware for our system, the software that is installed on that hardware and middle ware used to connect the disparate machine to another.

As we mention our project is web based system so there is different protocols that used to transfer data from one object to another object in the system such as file transfer protocol (FTP) and hypertext transfer protocol (HTTP).

Client interface

Web based server

Data base server

Client interface

Tourist site management

anagement

Car

Operator account management

Hotel

Data base

Persistence

HTTP

RMI

Finger 21: Deployment Diagram

## 4.5 Persistence Data Management

In this section we have to describe about how data store in the system and data manages infrastructure required form. The system will use mysql database server to storing data. This will allow the data base to easy integrate with and accessed by the rest of the system. The data base will retain tourist information (user name, password) and configuration of data such as authorized admin. Each of those item stores in separate table.

As we have to describe the system will be table which store in mysql server. Some these tables are;

* Register table: which store the tourist register information
* Operator table: which store operator account in the system
* Car table: which store car information
* Schedule table: which store information about the schedule
* operator table: which contain user manager information
* Hotel table: a table which contain about hotel information
* Tourist site table : a table which contain tourist site about information

## 4.6 Access Control and Security

Many levels of security protect sensitive document and file from unauthorized viewers. Each user has their security access levels and each document have sensitive level. Depending on the access level of the user they will see only the list of document that is appropriate for their security access level. General all user have their own user name and password to control security access level and document sensitive level.

The system accessed by difference account levels:

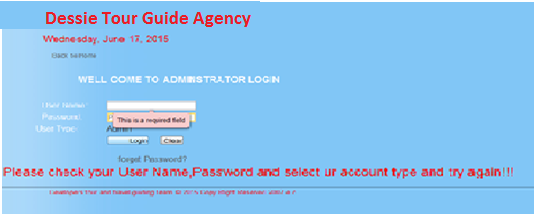
* The admin has been guaranteed to update operator account, delete operator account, create operator account, manage tourist site, login the system, add car, delete car, update car, interact database hotel and assign operator.
* The operator has been guaranteed to view the tourist site reservation, view car reservation, view hotel reservation, login the system and add operator account.
* The tourist has been guaranteed to register, reservation tourist site, reservation car, reservation hotel, update schedule and login the system.

## 4.7 User Interface Design

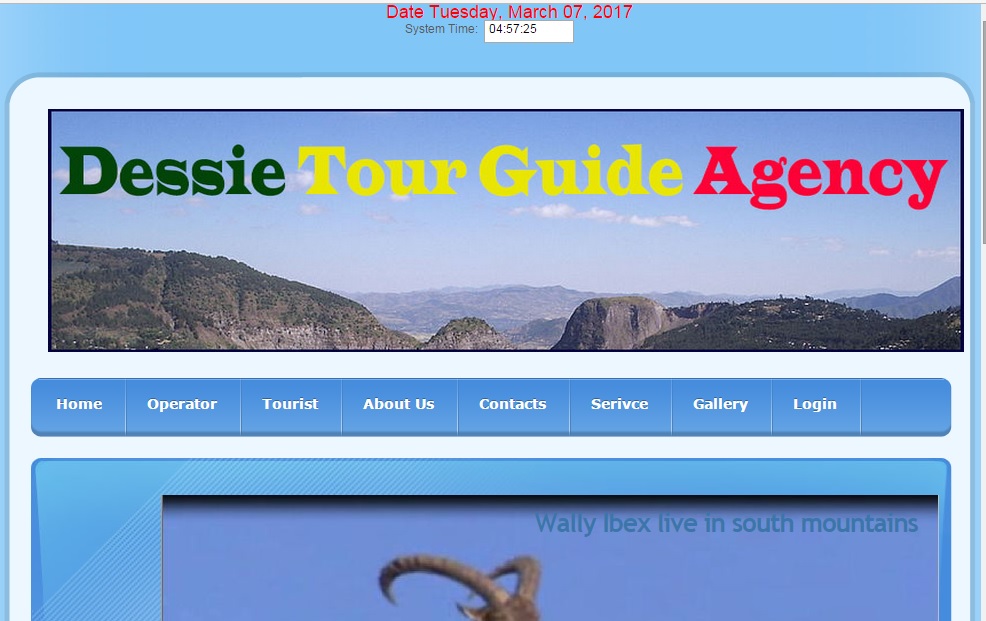
Tour and travel guiding system objectives is to provide user with well-organized user interface that could help users to spend less time to understand how they can use the system and what the system can offer to them. Another objective is to avoid user to spend much time entering the time access the system for each control unit

The following user interface is the work space or home page for the Tour and travel guiding system. User can select any user type from the menu or from links by clicking on them.

1. Login pages
2. The following User interface shows error message "Please fill out this field" when a user clicks login Button or press enter without entering user name and password. And the message, “please check user name, password and select your account type and try again” is shown when user entered user name, password or account type is wrong.

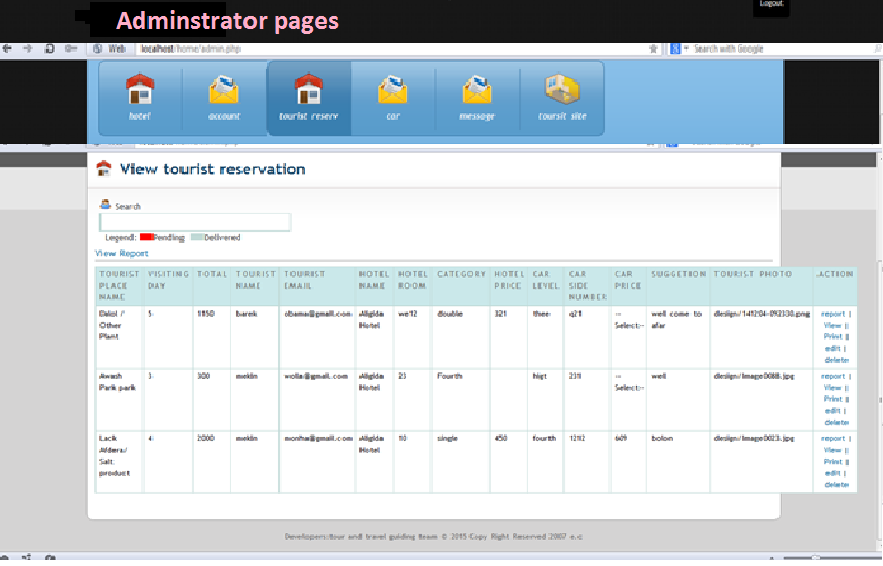


Finger 22: user interfaces login form

1. **home pages**

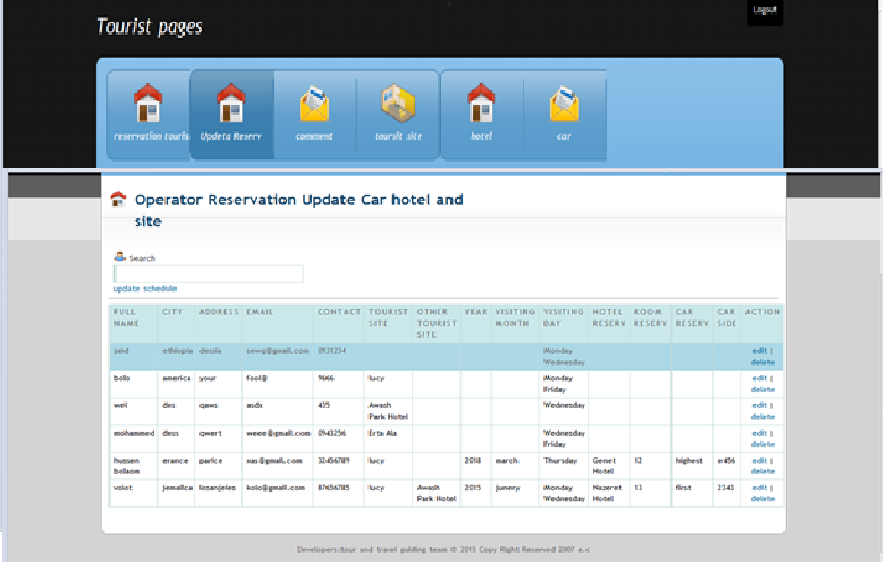
Finger 23: user interfaces home pages

1. **admin pages**

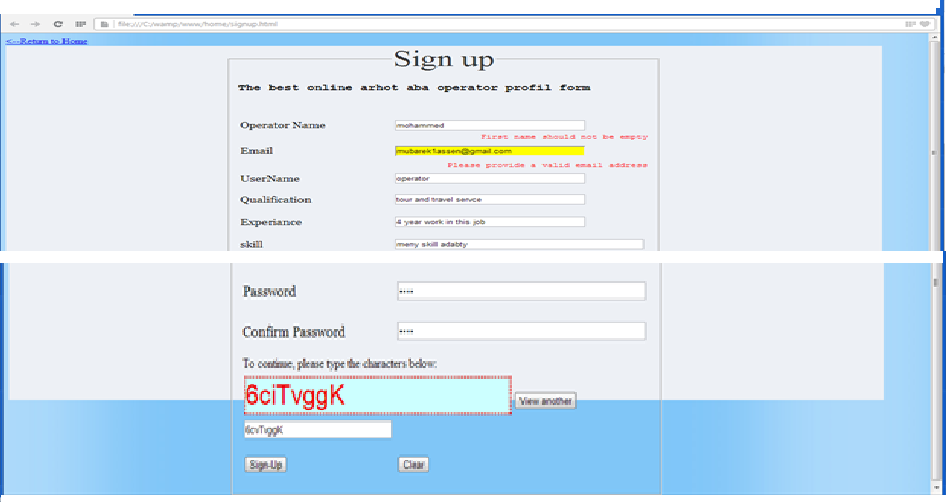
****

Finger 24: user interfaces administrator pages

1. **tourist pages**

****

Finger 25: user interfaces tourist pages

1. **Operator Profile form**

## Finger 26: user interfaces operator pages

## CHAPTER FIVE

## 5 Implementation and Testing

## 5.1 Implementation Overview

Implementation is the part of the process where software engineers actually program the code for the project. Implementation of a project is the step where all the proper planned activities are put into action. Usually project implementation process involves register tourist, manages tourist site, generate report register tourist, manages car, reservation tourist site, and register operator profile. Project admin and sometimes project team members are committed to controlling and monitoring project implementation process.

## 5.1.1Objective of the Implementation

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a probability of finding an as yet undiscovered error. A successful test is one that uncovers an as yet undiscovered error. Our Objective is to design test processes that systematically uncover different classes of errors and do so with minimum amount of time and effort.

## 5.1.2 Constraint of Implementation

In the process of implementation this project there were some constraints and challenges that limits to do what you went had been faced.

* Software constraint:-to develop this phase Wamp server, Note pad++ or Dreamweaver 8 are sometime corrupt.
* Hardware constraint:-to complete this project hardware part suddenly stops their application like muse, keyboard, socket and running part.
* Date base connectivity also one of constraint to running program.
* Lack internet access

## 5.1.3 Tools and Environment

PHP and MySQL run native on every popular flavor of UNIX (Including Mac OSx) and Windows. A huge percentage of the world’s HTTP servers run on one of these two classes of operating systems. PHP is compatible with the three leading Web servers: Apache HTTP Server for UNIX and Windows, Microsoft Internet Information Server, and Netscape Enterprise Server. We don’t require specific Web server compatibility with MySQL, since PHP will handle all the works for us.

## 5.1.3.1. Hardware components

For the early employment of system, a single machine is required. The recommended minimum system should have 3GHz Processor, 2GB RAM, and disk space required 250GB. For a usual, full deployment of this system, the minimum hardware environment should have been used.

## 5.1.3.2 Software Components

For the System implementation the following software are used.

* Operating system:- Window 7 and Windows 8, Windows 10
* Modeling Software: - Microsoft Visio2013.
* Programming language:- Php, Java script, Jquery
* Text Editors:- Macromedia Dreamweaver
* As Server Software:- Wamp server V. 2.1
* Database Software :- MYSQL 5.0
* Antivirus:- Avast
* Document tools:- MS word.

## 5.1.3.3 Environments

The following software application must be available on the planned hardware environment before installing the system. ***Browsers***: (One requires Client- side):

* Internet Explorer enabled.

***Web application Server:*** (One requires- Server side):

* Apache HTTP Server

***Relational Database:*** (One requires- Server side):

* MySQL Server

## 5.2 Sample Code and Description

Login form of the system is one privilege of the admin, tourist and operator uses to open its all privilege in all over the system. This means when admin try to enter in to the system he/she must have login first. The admin also have a right to give this privilege for the operator of the office. This means the operator also must have login first to open its privileges. The login account of operator is assigned by the administrator of the system. And this login form has an authentication if operator enters unknown password and he/she couldn’t access if the password is not correct.

## Source code of login Admin

<?php

if (isset($\_POST['Login']))

{

function clean($str) { $str = @trim($str);

if (get\_magic\_quotes\_gpc()

{ $str = stripslashes($str);

} return mysql\_real\_escape\_string($str);

}

$user\_name=clean($\_POST['user\_name']); $password=clean($\_POST['password']); $user\_type=clean($\_POST['user\_type']);

$login\_query=mysql\_query("select \* from users where user\_name='$user\_name' and password='$password' and user\_type='Admin'"); $login\_query1=mysql\_query("select \* from users where user\_name='$user\_name' and password='$password' and user\_type='tourist'");

$count=mysql\_num\_rows($login\_query); // this is 4 admin

$count1=mysql\_num\_rows($login\_query1); // this is 4 tourist

$row=mysql\_fetch\_array($login\_query);

$row1=mysql\_fetch\_array($login\_query1);

$f=$row['first\_name'];

$l=$row['last\_name'];

$user\_type=$row['user\_type'];

if ($count == 1)

{

$logout\_query=mysql\_query("select \* from users where user\_id='".$row['user\_id']."'");

$row=mysql\_fetch\_array($logout\_query);

$f=$row['first\_name'];

$l=$row['last\_name'];

$user\_type=$row['user\_type'];

mysql\_query("INSERT INTO history (data,action,date,user)VALUES('$f $l', 'Login', NOW(),'$user\_type')")or die(mysql\_error()); session\_start();

session\_regenerate\_id();

$\_SESSION['id']=$row['user\_id'];

header('location:admin.php');

session\_write\_close();

exit();

}

else if ($count1 == 1)

{ session\_start();

$\_SESSION['id']=$row2['user\_id'];

$logout\_query=mysql\_query("select \* from users where user\_id='".$row['user\_id']."'");

$row=mysql\_fetch\_array($logout\_query);

$f=$row['first\_name'];

$l=$row['last\_name'];

$user\_type=$row['user\_type']; mysql\_query("INSERT INTO history (data,action,date,user)VALUES('$f $l', 'Login', NOW(),'$user\_type')")or die(mysql\_error()); session\_start();

session\_regenerate\_id();

$\_SESSION['id']=$row['user\_id'];

header('location:admin.php');

session\_write\_close();

exit();

} else

{

?> <div class="alert alert-error"><button class="close" data-dismiss="alert">×</button> Please check your User Name,Password and select your login account types and try again!!!

</div>

<?php }

}

?>

**Sample code for checking whether the field time**

<SCRIPT language=Javascript>

vartimee;

functionstopClock(){

clearTimeout(timee);

}

functionyourClock(){

varnd = new Date();

var h, m, s;

var time=" ";

h = nd.getHours();

m = nd.getMinutes();

s = nd.getSeconds();

if (s <=9) s="0" + s;

if (m <=9) m="0" + m;

if (h <=9) h="0" + h;

time+=h+":"+m+":"+s;

document.the\_clock.the\_time.value=time;

timee=setTimeout("yourClock()",1000);

}</SCRIPT>

## 5.3 Testing Overview

Final phase of implementation is testing. Testing is a process to show the correctness of the program. Testing is checking of the system workability in an attempt to discover errors and avoiding such errors from the system. In this the group members tested the entire system as a whole with all forms, code, modules. In this we tested all the functionalities in the System. All errors in the forms, functions, modules have been tested.

## 5.3.1 Test by scope

To make sure that the system has completed the intended functionalities. We try to test the system using all available tests: - Unit Test and Integration Test.

## 5.3.1.1 Unit test

The act of testing small components of a system to ensure they work While implementing the code, all small components such as functions, conditional statements, loops and others tested. In computer programming, Unit testing is a procedure used to validate that individual units of source code are working properly. So In this test the team try to test each module individually but not integrate the whole system. It focuses verification efforts even in the smallest unit of software design in each module. This is also known as ―Module Testing.

The testing is carried out in the programming style itself. In this testing each ,module is focused to work satisfactorily as regard to the expected output from the module .There are some validation checks for the fields.

Example Error handling login pages

## 5.3.1.2 Integration Test

This is also called component testing, addresses the issue of whether the classes in the system or components of the system work together properly. If they all work individually, they should work when we put them together. The problem of course is putting them together “This can be done in two ways:

**Top down integration*:*** Modules are integrated by moving downwards through the control hierarchy, beginning with main control module are incorporated into the structure in either a depth first or breadth first manner.

**Bottom up integration*:*** It begins with construction and testing with atomic modules i.e. modules at the lowest level of the program structure. Because modules are integrated from the bottom up, processing required for the modules subordinate to a given level is always available and the need of stubs is eliminated.

## 5.3.2 Testing by requirements

System test insures that the entire integrated software system meets requirements. It tests a configuration to insure known and predictable results. System testing is based on process description and flows, emphasizing pre-driven process links and integration points.

In essence system testing is not about checking the individual parts of design, but about checking the system as a whole. In effect it is one giant component.

System testing insures the following have been met correctly. They are:

* Functional requirements
* Non Functional requirements such as
* **Performance:-**Are the performance criteria met?

If our computer performance is high then the system test will show you that the performance requirement satisfied.

* **Volume:-**Can large volumes of information are handled?

Large volume of information can be handled in our system.

* **Documentation:-**Is the documentation usable for the system?

The documentation can be used as a guide for the user of the system.

* **Robustness:-**Does the system remain stable under adverse circumstances?

Is a system able to hold different obstacles?

## 5.3.3 Test Implementation (Validation testing)

It is the final step of testing**.** In this the team members tests the entire system as a whole with all forms, code, modules.

* **User name checking:**

When unauthorized users are try to access to the system displays an alert message “please check user name, password and select your account type and try again”.

* **Password checking:**

When unauthorized users are try to access to the system displays an alert message “please check user name, password and select your account type and try again”.

* **user type validations:**

Accepts only validate data .If you insert invalidate data the system display alert message “please check user name, password and select your account type and try again”.

* **Null input:**

Allows the system to accept the needed data rather than null. Else display an error message box. “enter the required”.

## CHAPTER SIX

## 6 User Manual

User manual is document describing the user interface of the system such that a user unfamiliar with the system can use it. It states about the operation information of the system. This part will help the user to develop an ability to read and understand the considerable information presented in the software. We have to use difference method to access the system

1. Run the wamp server from the Desktop or Start up Menu if it’s not activated then Click on start - >wamp server
2. From the taskbar check as the wamp server is activated, it should be green to be activate then click on it and select ->localhost
3. The browser lists all available projects under "Your Projects" then select "unity" directory.
4. The browser will display the home page of the system, the you know what to do.

## 6.1 Installation guide

Since the project is online, there is no need to install it on a particular machine rather it will be hosted on a server.

After installing wamp server software but its recommended to have both Macromedia Dreamweaver software's too and do the following procedures. Those procedures are:-

* First make sure that where the wamp server is installed on your computer the default is "c: /wamp" folder then
* Copy the folder home you take from the Developing team to your c:/wamp/www folder
* Activate wamp server
* click on wamp server and select local host; then the default browser take your the local host home page
* Next you have to insert all the database information to your server so
* Click on wamp server select phpMyAdmn
* Create database named "tours" by type the name on the create database box and press create button, then it will inform you that the database has been created.
* open the database you created by clicking on it.
* select "import " from the command listed
* Click on choose file button and Brows the file from home folder named "tours.sql" and click on open then click on go
* The server will insert all information and list all the tables
* Now you can access the system freely.
* Finally, log in with your account for stat up use this one "user name: admin" and password: admin
* Installation is finished.

## CHAPTER SEVEN

## 7. Conclusion and Recommendation

## 7.1 Conclusion

The main purpose of this project is to establish a long lasting and effective communication between different users and also to introduce various features regarding Dessie tour guide agency.

An effort has been made to study tour and travel guiding system as partial fulfillment of BSC degree in information system. In doing the study the team has tried to follow object oriented system analysis and design methodology.

Since the success and failure of any system depends on gathering the right information through different fact-finding techniques and user involvements, the team has made the better effort to gather requirements. After a detail review and study of the existing system of tour and travel guiding system models have been designed to reflect the new system that are supposed to solve problems.

In order to solve different problems existed the team has tried to propose a solution that at least reduce the existed problems and model the proposed system using different tools and methodologies. The team believe the different tools and techniques has helped us a lot in capturing real user requirements and model the right system for the users for their day to day transactions. Thus it should have the precedence in know-how and experience in collecting, processing and utilizing information.

This project also focuses on online tourist access necessary information about Dessie tour guide agency and their tourist site, but it has a little bit difficult to implement.

## 7.2 Recommendation

Since we are now living in a world that is led by technology and technology results, we need more and more applications to familiarize ourselves and also come up with the fast advancing technology. Thus, as we are beginner website developers, we recommend that other website designers, beginners or professionals, to create more dynamic pages that are very user friendly, more secure and also introduce the community as a whole to be familiar with the current technology**.** Online payment is difficult to the security purpose; we hope that this project will create some initiation for those people who wants to develop online payment.

## Reference

* An Object-Oriented Approach to Software Development for Parallel Processing Systems
* Bernd bruegge and allen H.Dutoti, Object-Oriented Software Engineering UsingUMLPattern Java,PrenticH
* Dessie Tourism Association
* Ian Somerville “Software engineering” Pearson Education Limited 7th edition 2004
* National regional culture and tourism bureau for tourist guide book
* Object Oriented And Classical Software Engineering 8th Edition
* Object Oriented And Classical Software Engineering 8th Edition V413HAV
* Prentice Object Oriented Software Engineering Using UML Patterns and Java 3rd 2012 (1)
* Prentice Object Oriented Software Engineering Using UML Patterns and Java 3rd 2012(2)
* Stephen R. Schema “Object-Oriented Analysis and classical software engineering” McGraw hill publisher 6th edition 2000