**Project**

**On**

**Tour Management System**

****

Dawa Tenzing Lama

00172886

Email: tenzing0070@gmail.com

Computing Project

Level 5 Diploma in Computing

Softwarica College of IT & E-Commerce

Kathmandu, Nepal

*April 2019*

Submitted to: Kiran Rana

Contents

[**Table of Figures** 4](#_Toc5145190)

[**Chapter 1 Introduction** 5](#_Toc5145191)

[**Introduction** 5](#_Toc5145192)

[**Background of the project** 6](#_Toc5145193)

[**Aims** 6](#_Toc5145194)

[**Objectives** 7](#_Toc5145195)

[**Chapter 2 ANALYSIS** 8](#_Toc5145196)

[**Introduction to Analysis** 8](#_Toc5145197)

[**Requirement Analysis** 9](#_Toc5145198)

[**Functional and Non-Functional** 9](#_Toc5145199)

[**Functional requirements:** 9](#_Toc5145200)

[**Non-Functional Requirements:** 10](#_Toc5145201)

[**Feasibility study** 11](#_Toc5145202)

[**Advantages:** 12](#_Toc5145203)

[**Disadvantages:** 12](#_Toc5145204)

[**Use-Case Diagram** 12](#_Toc5145205)

[**Initial Class Diagram** 14](#_Toc5145206)

[**Chapter 3 DESIGN** 17](#_Toc5145207)

[**Design Introduction** 17](#_Toc5145208)

[**Structural Design** 18](#_Toc5145209)

[**Class Diagram** 18](#_Toc5145210)

[**Class Diagram** 18](#_Toc5145211)

[**Behavioral Design** 20](#_Toc5145212)

[**Activity Diagram** 20](#_Toc5145213)

[Admin Activity Diagram 21](#_Toc5145214)

[**Generating receipt activity diagram** 23](#_Toc5145215)

[**Package reservation activity diagram** 25](#_Toc5145216)

[**Sequence Diagram** 27](#_Toc5145217)

[**Login Sequence Diagram** 27](#_Toc5145218)

[**Admin Sequence Diagram** 29](#_Toc5145219)

[**User Sequence Diagram.** 30](#_Toc5145220)

[**Database Design** 32](#_Toc5145221)

[**Entity Relationship Diagram** 32](#_Toc5145222)

[**Chapter 4 UI DESIGN** 34](#_Toc5145223)

[**UI Design** 35](#_Toc5145224)

[**Registration** 35](#_Toc5145225)

[**Login Form** 36](#_Toc5145226)

[**Home Page** 37](#_Toc5145227)

[**Gallery** 38](#_Toc5145228)

[**Package** 39](#_Toc5145229)

[**Booking** 40](#_Toc5145230)

[**Contact** 41](#_Toc5145231)

[**Admin Dashboard** 42](#_Toc5145232)

[**Package Booked Information** 43](#_Toc5145233)

[**Admin package CRUD** 43](#_Toc5145234)

[**Admin Gallery Function** 44](#_Toc5145235)

[**User information** 44](#_Toc5145236)

[**Chapter 5 TESTING** 45](#_Toc5145237)

[**Introduction** 46](#_Toc5145238)

[**Principles of Testing** 46](#_Toc5145239)

[**Black box Testing** 46](#_Toc5145240)

[**Unit Testing** 48](#_Toc5145241)

[**Conclusion** 49](#_Toc5145242)

# **Table of Figures**

[Figure 1Usecase Diagram 13](#_Toc5145115)

[Figure 2Initial class diagram 15](#_Toc5145116)

[Figure 3Class diagram 19](#_Toc5145117)

[Figure 4Admin activity. 21](#_Toc5145118)

[Figure 5Generating receipt 23](#_Toc5145119)

[Figure 6Package reservation 25](#_Toc5145120)

[Figure 7Login sequence diagram 27](#_Toc5145121)

[Figure 8Admin sequence diagram 29](#_Toc5145122)

[Figure 9User sequence diagram 30](#_Toc5145123)

[Figure 10ER diagram 32](#_Toc5145124)

[Figure 11Registration 35](#_Toc5145125)

[Figure 12Login 36](#_Toc5145126)

[Figure 13Homepage 37](#_Toc5145127)

[Figure 14Gallery 38](#_Toc5145128)

[Figure 15Package 39](#_Toc5145129)

[Figure 16Package details 39](#_Toc5145130)

[Figure 17Book package 40](#_Toc5145131)

[Figure 18Comment box 41](#_Toc5145132)

[Figure 19Admin page 42](#_Toc5145133)

[Figure 20Admin package booked information. 43](#_Toc5145134)

[Figure 21Admin Package CRUD 43](#_Toc5145135)

[Figure 22Admin Gallery Function. 44](#_Toc5145136)

[Figure 23User information. 44](#_Toc5145137)

# **Chapter 1 Introduction**

## **Introduction**

Tour management is web-based application which is designed to automate the travel process to the customers through online facilities for travelling, easy access to the relevant information and necessary travel booking and arrangement.

The complete information about the project has been highlighted in this project proposal. This application is going to add easiness for those users using such type of application and meet their satisfaction. This application will have various features for fulfilling the user requirement.

## **Background of the project**

Touring is a part of a life as which can sometimes lead to uncertain situation. So, in order to overcome that problem, this project will help users to get all the required information about the schedule and safe routes for their trip.

Solving this problem can uplift the range of tourism to go higher and increase the revenue too. Through this, customers can be able to utilized various benefits and resources such as safe routes, booking, destination information, reviews about the places, environments etc.

The project will be developed using Object Oriented PHP. Different techniques and procedures will be implemented that will enhance the quality of application.

## **Aims**

* To help the organization to build a proper system to record the travelling information.
* To provide proper information of the travelling routes, destination, locals etc.
* To make desktop application and online web application to create communication and sharing of information.
* To develop database system to store the information of the travelers, places etc.

## **Objectives**

* To collect the exact number of data of the travelers of different parts of the world.
* To give a brief information of the different beautiful places.
* To make user friendly GUI system which can also be used by non-technical users (finding user experiences)

# **Chapter 2 ANALYSIS**

## **Introduction to Analysis**

Analyzing is the method of identifying and documenting of requirement of the proposed system. In the analyzing phrase, the first phase is to feasibility study after that we model the system in use case diagram and class diagram which is also known as system modelling.

## **Requirement Analysis**

## **Functional and Non-Functional**

Functional requirements simplify defines what a system is supposed to do. It specifies a particular behavior of function of the system when certain conditions are met.

Non-Functional requirements define how a system is supposed to be. It specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.

In the given table, the functional and non-functional requirements are identified and then they are prioritized with MoSCoW prioritization.

Index used in the table below are:

**F**=Functional requirements

**NF**=Non-Functional requirements

**M**=Must Have

**S**=Should Have

**C**=Could Have

## **Functional requirements:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Functional** | **Title** | **MoSCoW** | **Description** |
| **F(R1)** | Registration | **M** | User must registration to system. |
| **F(R2)** | Login | **M** | User must Login to System for booking packages. |
| **F(R3)** | Tour package option | **S** | Able to select different package option. |
| **F(R4)** | Package booking | **M** | Able to book package. |
| **F(R5)** | Routes information | **S** | To get the destination information. |
| **F(R6)** | Search package | **M** | Able to search package. |
| **F(R7)** | Generate and print receipt | **M** | Receipt must be generated after booking. |
| **F(R8)** | Change password | **M** | User can change password accordingly. |
| **F(R9)** | Online payment | **W** | Online transaction cannot be made. |
| **F(R10)** | Profile add, update, delete, view | **M** | Able to add, update, delete personal information. |
| **F(R11)** | Show package booked details | **M** | Can see package booking details |
| **F(R12)** | Package add, update, delete, view | **M** | Admin is able to add, delete, update, view package data. |
| **F(R13)** | Viewers Feedback | **M** | Can check viewers feedback |
| **F(R14)** | Booking History | **M** | Admin can check the booking history. |

## **Non-Functional Requirements:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Non - Functional** | **Requirements** | **MoSCoW** | **Description** |
| **NF 1** | Performance | **M** | System should be performed smoothly |
| **NF 2** | Usability | **M** | System should navigation easily and easy to use. |
| **NF 3** | Reliability | **C** | Precise and accurate information should give to system |
| **NF 4** | Interface | **C** | User interface should be easy and friendly. |
| **NF 5** | Portability | **S** | System should be run in every platform. |
| **NF 6** | Scalability | **C** | System should store more and more information easily |
| **NF 7** | Security | **M** | Make sure that System must secure and protected from unauthorized access. |
| **NF 8** | Website Backup | **S** | System should backup accordingly. |
| **NF 9** | Maintainability | **C** | Maintenance should be done in regular basic |
| **NF 10** | Privacy | **M** | System should keep user information private |

## **Feasibility study**

Feasibility study helps to define of idea which is following a project, legally and technically feasible and economically just able. It also helps to identify project is potentially successful or not.

## 

## **Advantages:**

* It helps to provide valued information details for the project.
* Through evaluation process of multiple factor, it has improved more success rates.

## **Disadvantages:**

* High price.
* Incorrect information.

## **Use-Case Diagram**

Use-case is a software and system engineering term that define how user can interact with system in order to accomplish a particular operation. A use case acts as a software modelling technique that implies function to be implemented. (Techopedia, 2018)

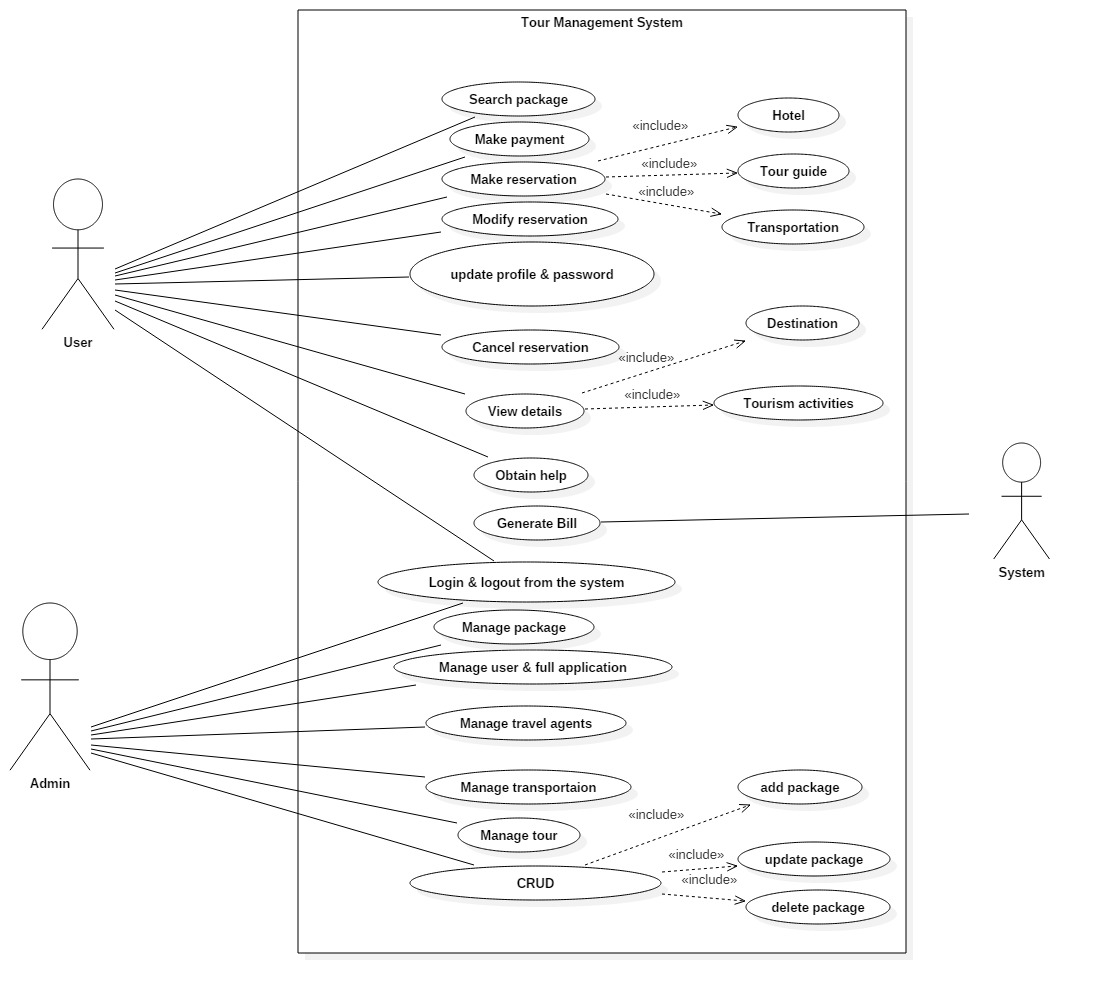
****

Figure 1Usecase Diagram

**Justification**

* For gathering the system requirements and actors and to collect the functional requirements of a system.
* To define the set of actions that system can execute by collaborating with one or more external users.
* To specify the events of a system and their flows.
* In above given diagram, three different actors i.e. (User, Admin, System) has been displayed. User is unregistering customer for the system, second one is admin who manages whole system and the third is the system.
* First actor i.e. user who can register into the system by filling up registration form become the registered customer. After registration, he/she can do all things that registered customer can do to the system.
* Second actor i.e. admin who is able to control and access whole system. Admin is able to manage, add, delete, update, view the customer and package information.
* Third actor i.e. system which helps to generate bill.

## **Initial Class Diagram**

A class diagram is explanation/report of relationship in between different multiple classes. It is a static diagram that displays a static view of system and shows a collection of interface, classes and association. It also defines the attributes and operations of class.

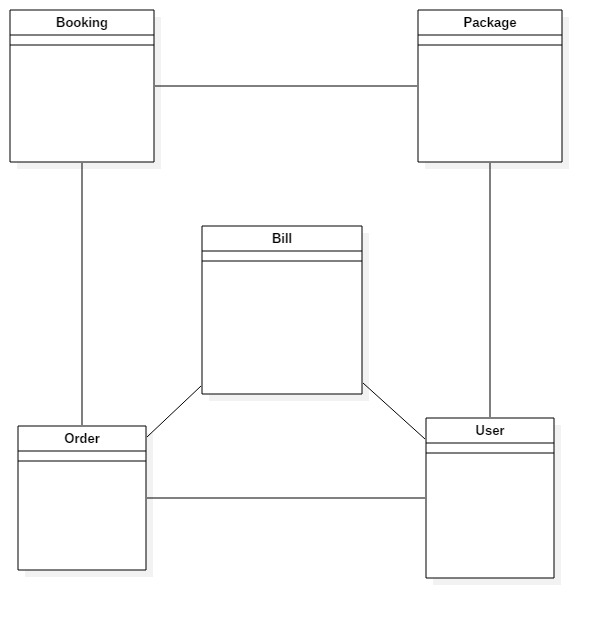


Figure 2Initial class diagram

**Justification**

* In the class diagram, classes are represented with boxes that contain three compartments:
* The top compartment contains the name of the class, printed in bold and centered, and the first letter is capitalized.
* The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.
* The bottom compartment contains the operations, the class can execute. They are also left-aligned and the first letter is lowercase
* Class diagram are simple and fast to read. Forward and reverse engineering.
* To model the static view of an application, giving a sense of orientation.
* They provide detailed insight into the structure of our systems, describing responsibilities of a system.

# **Chapter 3 DESIGN**

## **Design Introduction**

It is a detailed document that provides full information about a going to be developed product of process. Its main aims are to ensure that the development product must meet the users’ requirements. There are three different types of design phrase i.e. structural design, behavioral design and database design.

## **Structural Design**

It is architectural of the system that emphasis on the class, object and method of the system.

## **Class Diagram**

It is a type of static view of the system that describes the whole structure by showing classes, attributes, method or operation and relationship between classes of the system. (Visual-Paradigm, 2019)

It is used to illustrate and create a functional diagram of the system classes and serves as a system development resource within the software development life cycle.

## **Class Diagram**

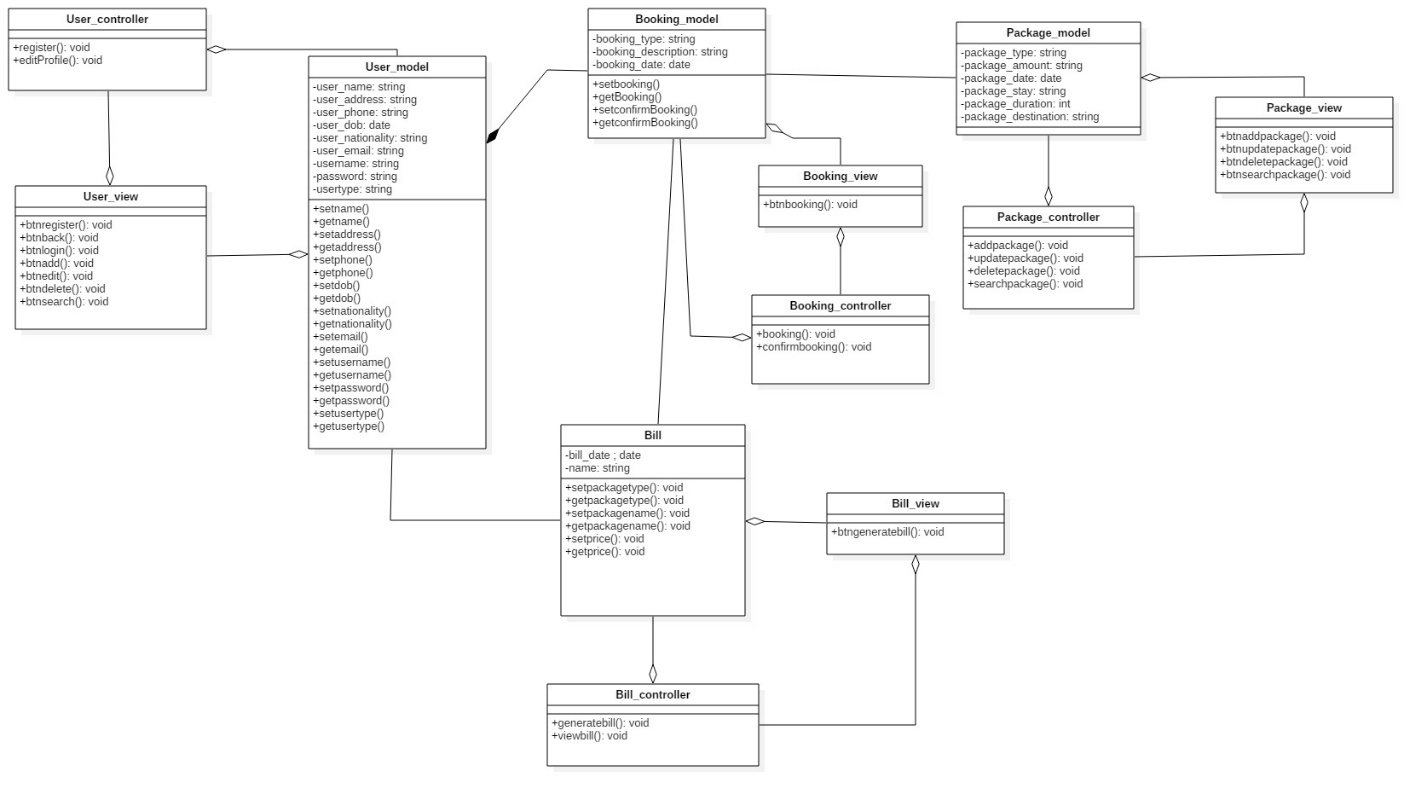


Figure 3Class diagram

**Justification**

* Displaying each class has each controller and the controller are directly linked with the database.
* Simple and easy to understand.
* It helps to model the static view of an application and describes the responsibilities of the system.
* It gives the detail insight into the structure of our systems but shows only the collaboration in between elements of the static view.
* For visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.
* Base for component and deployment diagrams.

## **Behavioral Design**

Concept that focuses on how a structure or a system, as viewed by the users, meets their needs and requirements.

## **Activity Diagram**

It is a UML diagram that describes the dynamic aspect as well as advanced version of flow chart which show logic of flow of one activity to another activity of the system. The activity can be described as an operation of the system. The flow can be sequential, branched or concurrent. It deals with all type of flow control by using different elements like fork, join etc.

## Admin Activity Diagram

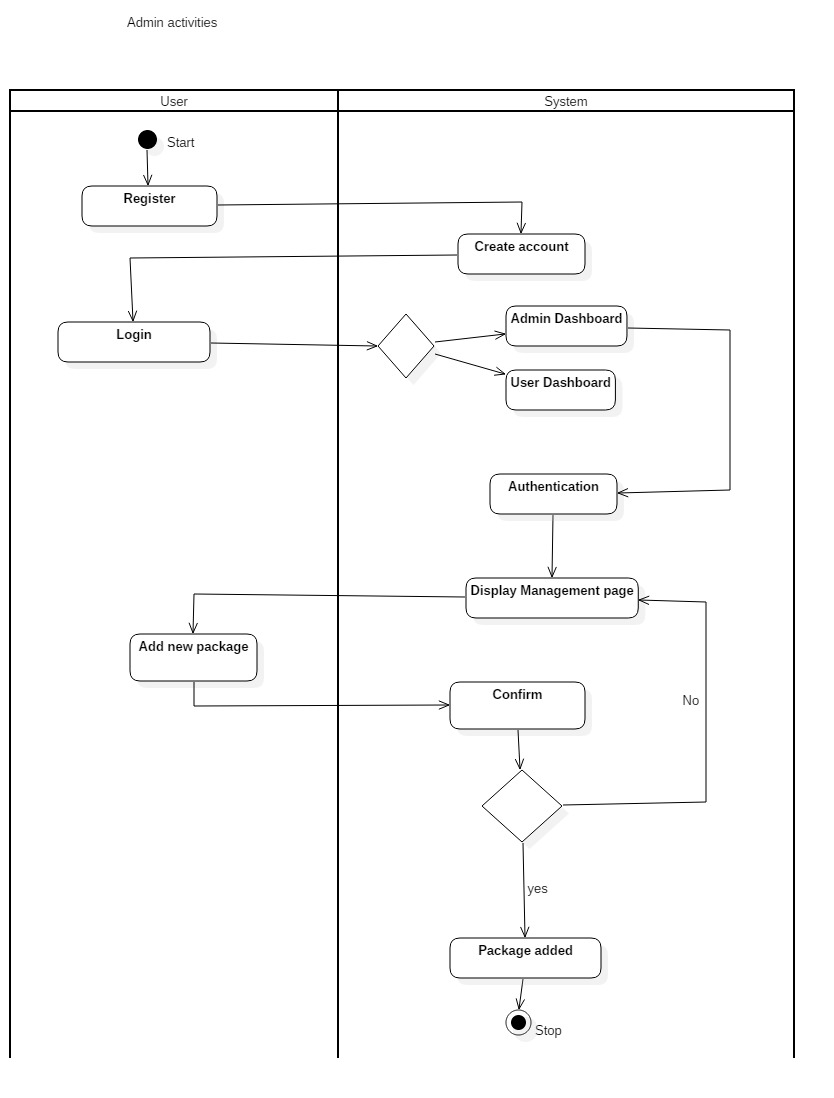


Figure 4Admin activity.

**Justification**

In the above given activity diagram, admin activities have been displayed. At first the user can register to create an account and login to the system. If the logged in user is valid as Admin then then the system will direct to Admin dashboard. In order to add new package in the system, the admin sends the command and system confirms the request. If the request is accepted, it creates the new package else it goes back to Admin dashboard.

* Activity diagram is basically a flow chart to represent the flow from one activity to another activity
* To describe the sequence from one activity to another.
* To describe the parallel, branched and concurrent flow of the system.
* The flow can be sequential, branched or concurrent. It deals with all type of flow control by using different elements like fork, join etc.
* Displays multiple conditions and actors within a work flow through the use of swim lanes.

## **Generating receipt activity diagram**

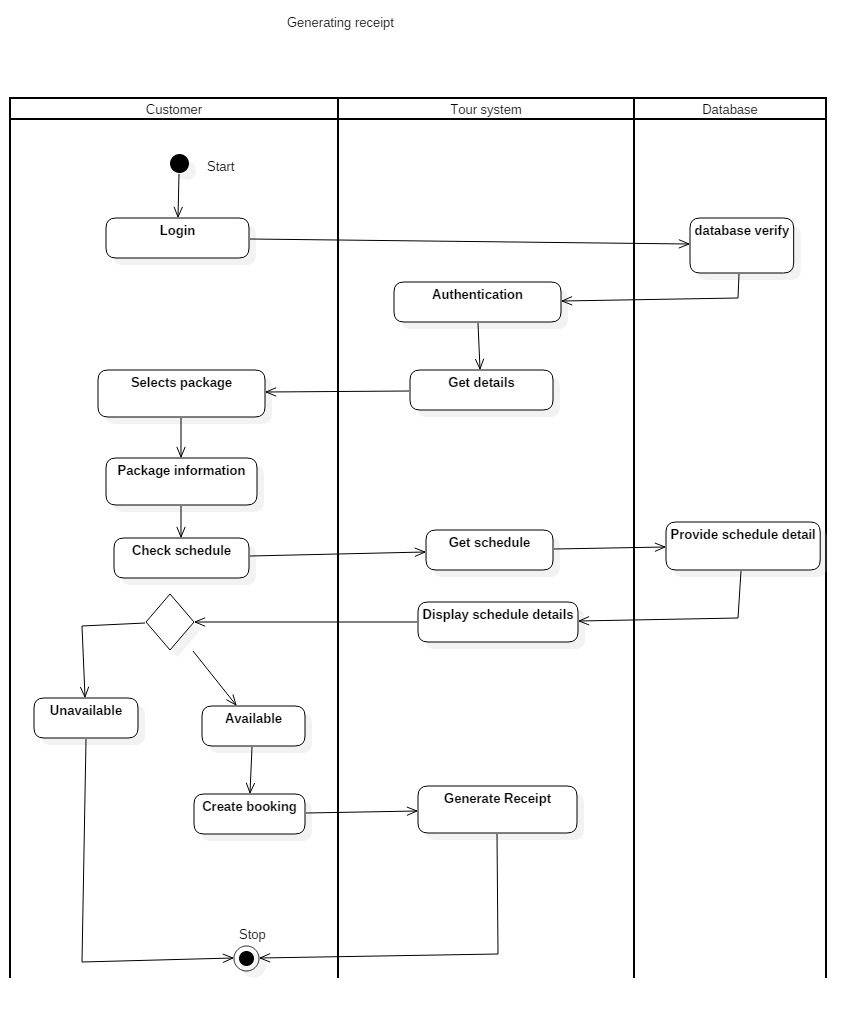


Figure 5Generating receipt

**Justification**

In the above given activity diagram, generating receipt activity has been displayed. The user login to the system where database verifies the registered user and enters into the page and able to retrieve details. The users select the appropriate package, checks schedule for the availability through the database system, where the information will be displayed in the page. If the package if unavailable it cancels the booking and if the package schedule is available then it creates booking for the package and the system generates the receipt of the appropriate package.

* The activity can be described as an operation of the system.
* Activity diagram is basically a flow chart to represent the flow from one activity to another activity
* To describe the sequence from one activity to another.
* To describe the parallel, branched and concurrent flow of the system.
* The flow can be sequential, branched or concurrent. It deals with all type of flow control by using different elements like fork, join etc.
* Displays multiple conditions and actors within a work flow through the use of swim lanes.

## **Package reservation activity diagram**

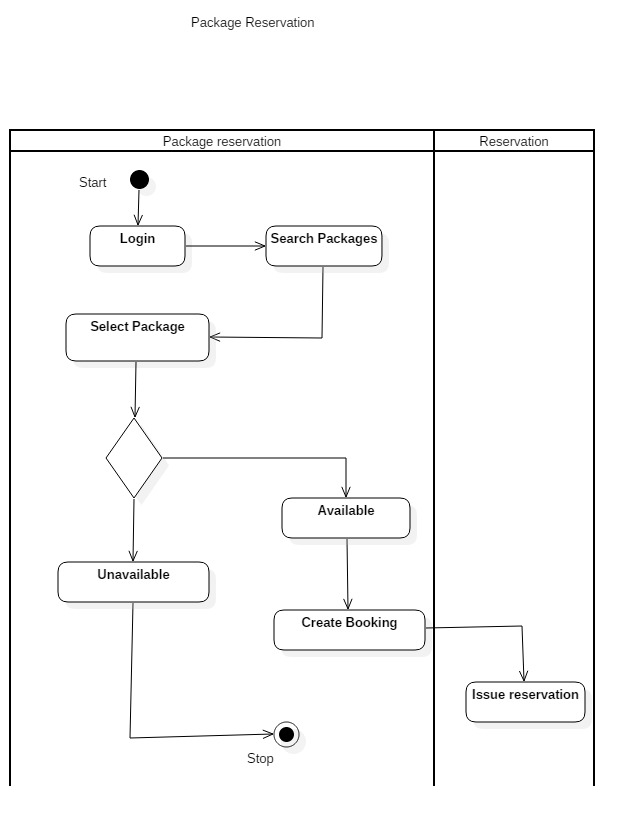


Figure 6Package reservation

**Justification**

In the above given activity diagram, package reservation activity has been displayed. User can login in to the system and search for packages accordingly. User can select the package and checks for availability. If the package is unavailable, the user gets the messages of unavailability and if the package is available, user can book the package and complete the reservation accordingly.

* The activity can be described as an operation of the system.
* Activity diagram is basically a flow chart to represent the flow from one activity to another activity
* To describe the sequence from one activity to another.
* To describe the parallel, branched and concurrent flow of the system.
* The flow can be sequential, branched or concurrent. It deals with all type of flow control by using different elements like fork, join etc.
* Displays multiple conditions and actors within a work flow through the use of swim lanes.

## **Sequence Diagram**

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

## **Login Sequence Diagram**

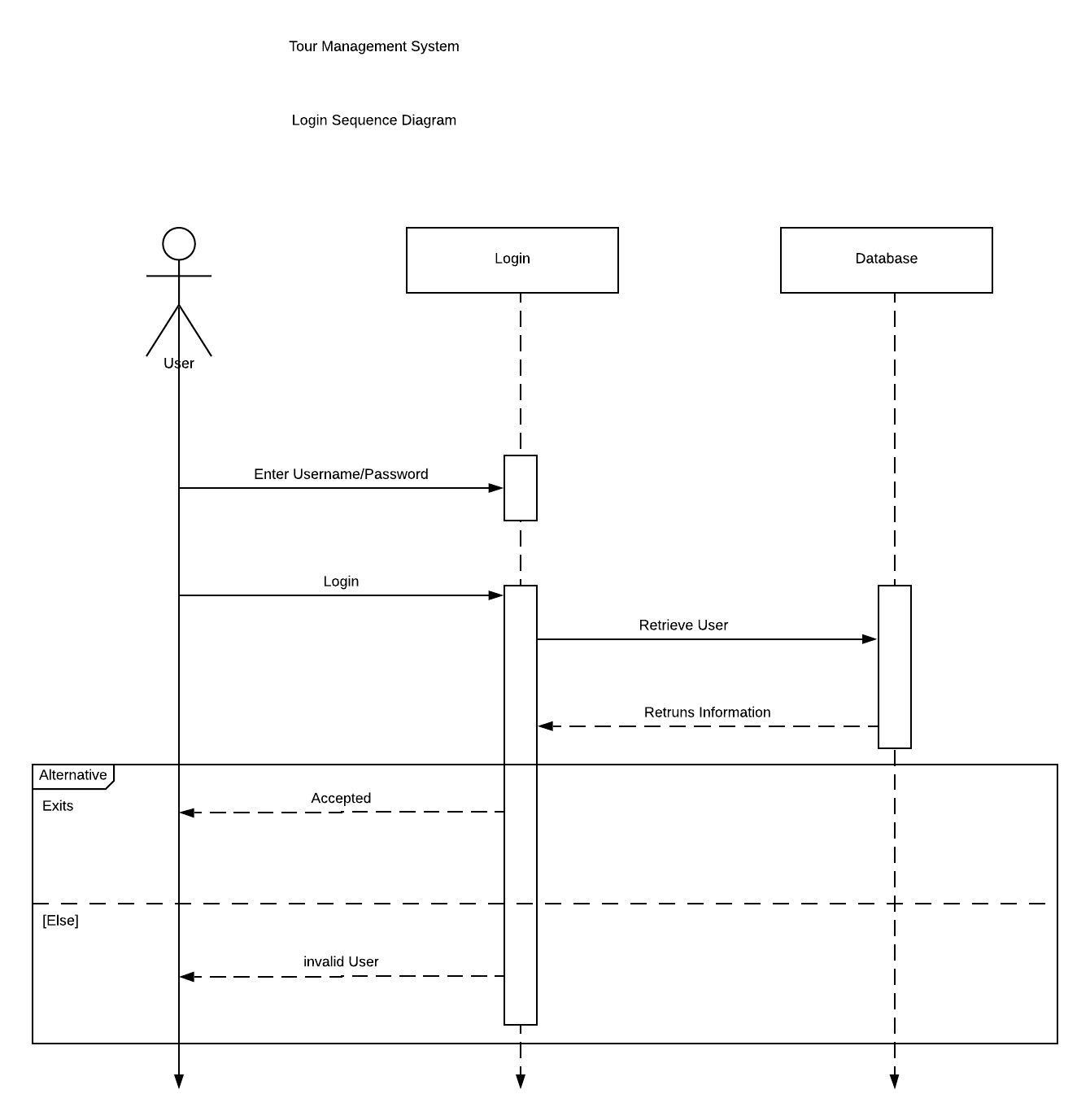


Figure 7Login sequence diagram

**Justification**

In the above diagram, login sequence diagram has been displayed. User enters the username and password where the database checks the information and give authorization of login the system accordingly.

* The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur.
* Easier to understand.
* UML specification is more sequence diagram centric.
* Allows reverse engineering.
* Help developers and business analyst to get a common understanding.

## **Admin Sequence Diagram**

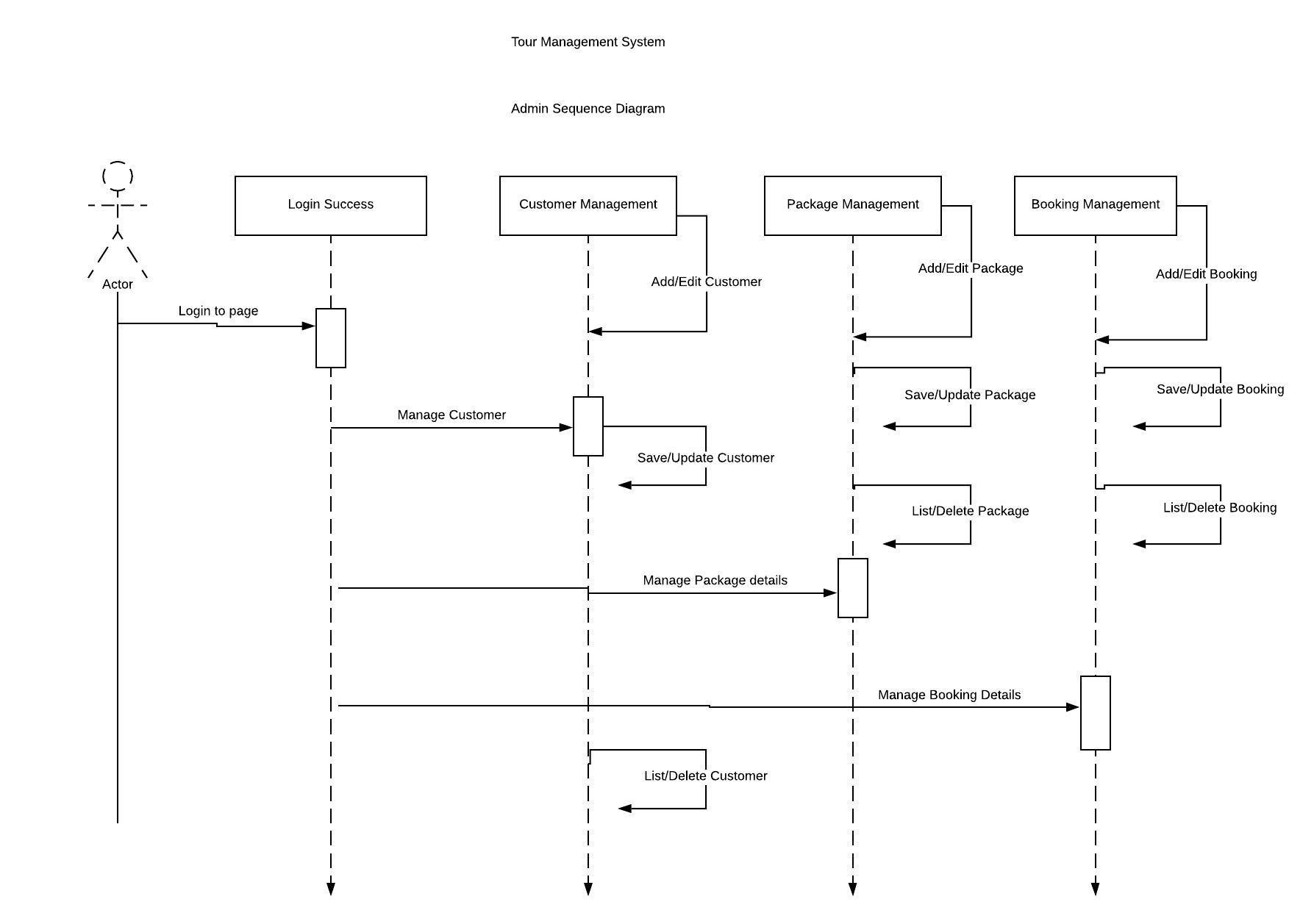
****

Figure 8Admin sequence diagram

**Justification**

In the above sequence diagram, admin CRUD function has been displayed.

* The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur.
* Easier to understand.
* UML specification is more sequence diagram centric.
* Allows reverse engineering.
* Help developers and business analyst to get a common understanding.

## **User Sequence Diagram.**

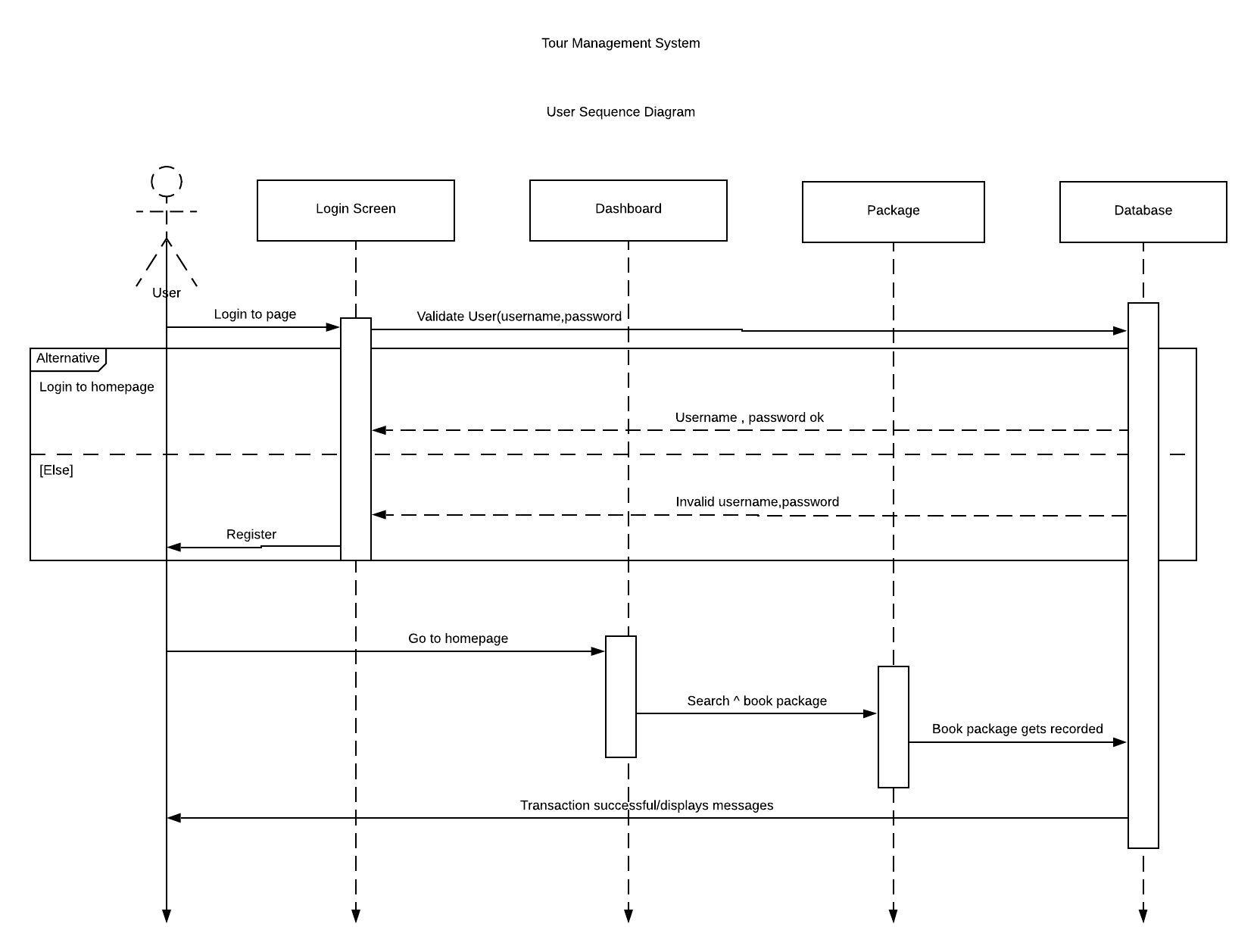


Figure 9User sequence diagram

**Justification**

The above sequence diagram shows the user booking activity. The user can login in to the system and able to select the package accordingly and book the package.

* The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur.
* Easier to understand.
* UML specification is more sequence diagram centric.
* Allows reverse engineering.
* Help developers and business analyst to get a common understanding.

## **Database Design**

It is the process of designing, developing and implementing the data for proposed system. Its main aim is to produce a logical and physical model for any system.

## **Entity Relationship Diagram**

An Entity Relationship Diagram also called as ERD is a type of flowchart that illustrates how entities relate with each other within system.

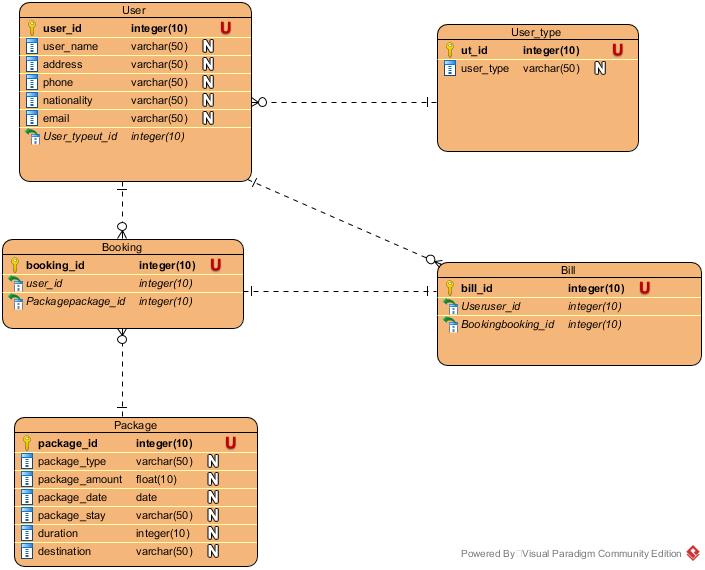


Figure 10ER diagram

**Justification**

* It helps to find out the mistakes and flaws of the design, and able to correct before executing the changes in database.
* Easily the entities can be located by visualizing the database schema, attributes can be viewed and helps to identify the relationship.
* Better visual representation and simple if we know relationship between entities and attributes.
* An effective communication tool for database designer.
* Conversion from E-R diagram to a network or hierarchical data model can· easily be accomplished.
* Clear understanding of the data structure and minimizes redundancy and other problems

# **Chapter 4 UI DESIGN**

## **UI Design**

User interface (UI) design is the process of making interfaces in software or computerized devices with a focus on looks or style. Designers aim to create designs users will find easy to use and pleasurable. UI design typically refers to graphical user interfaces but also includes others, such as voice-controlled ones.

## **Registration**

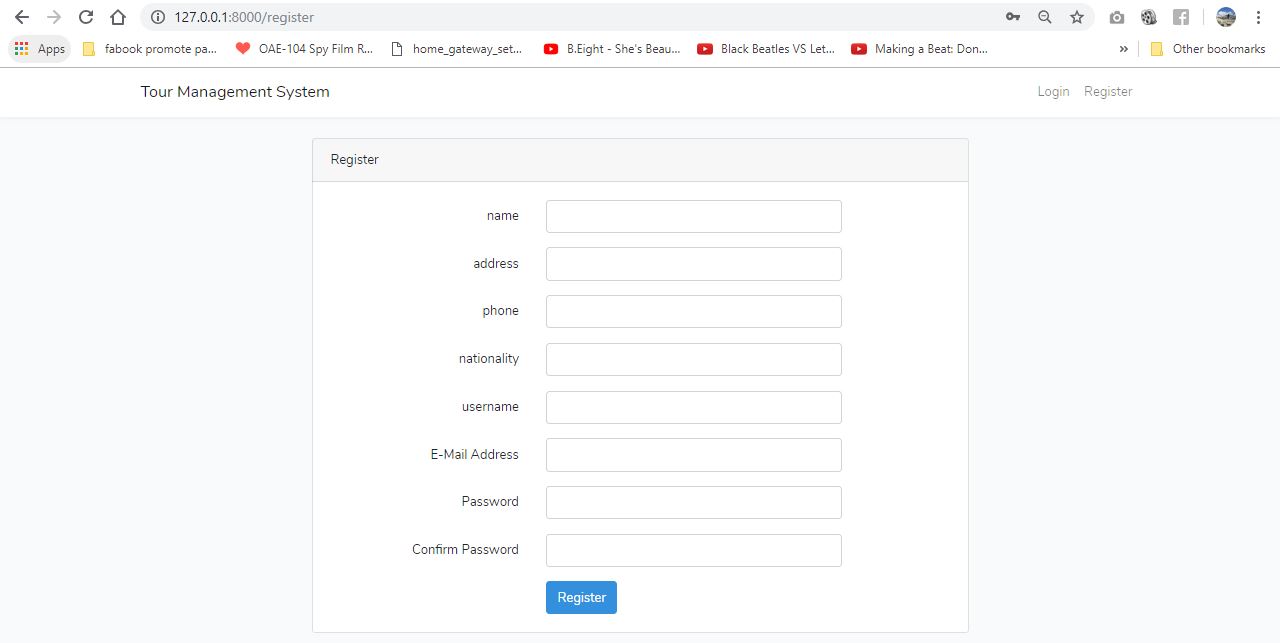


Figure 11Registration

## **Login Form**

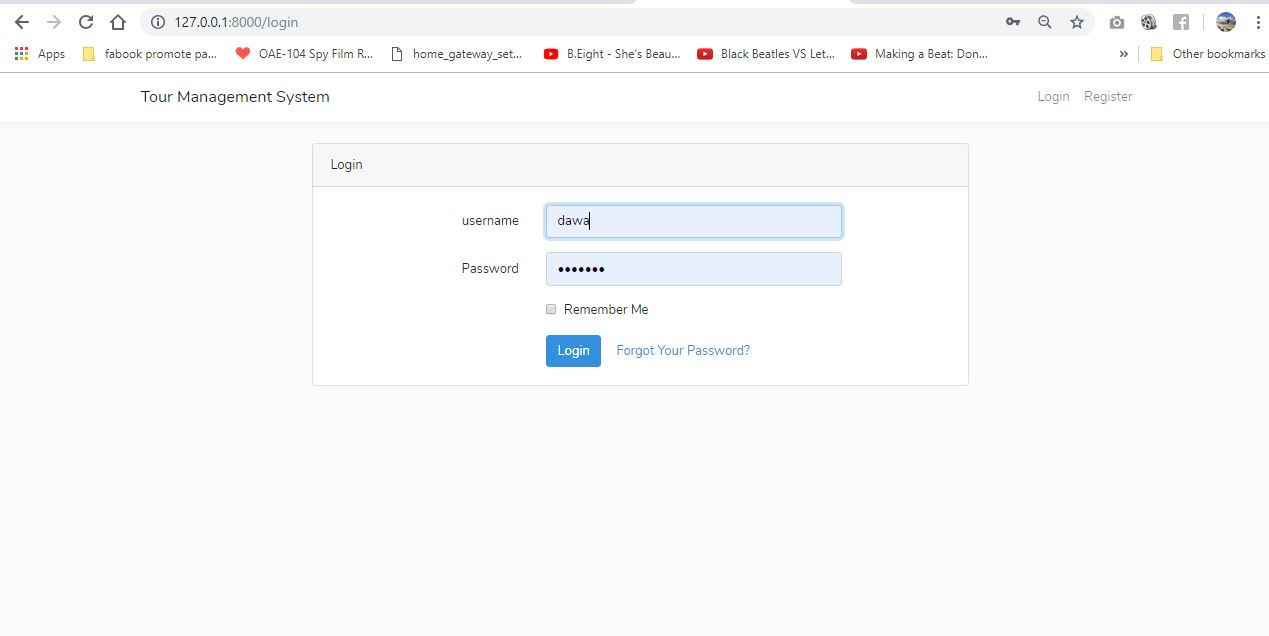


Figure 12Login

## **Home Page**

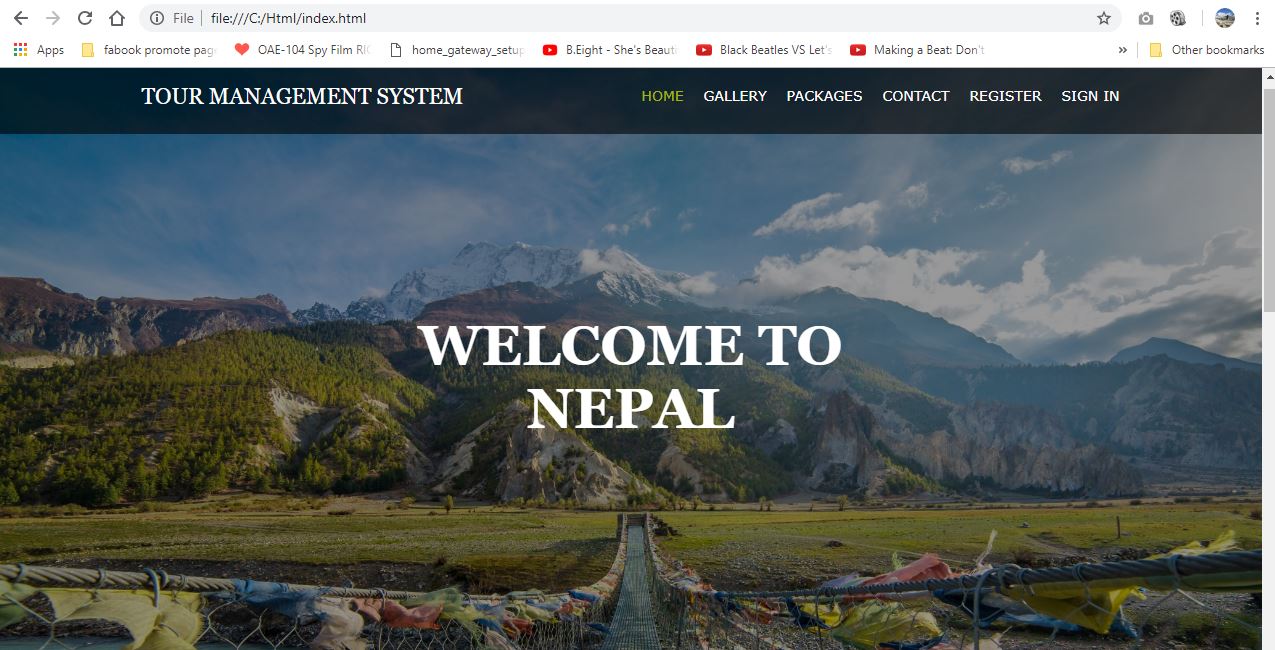


Figure 13Homepage

## **Gallery**

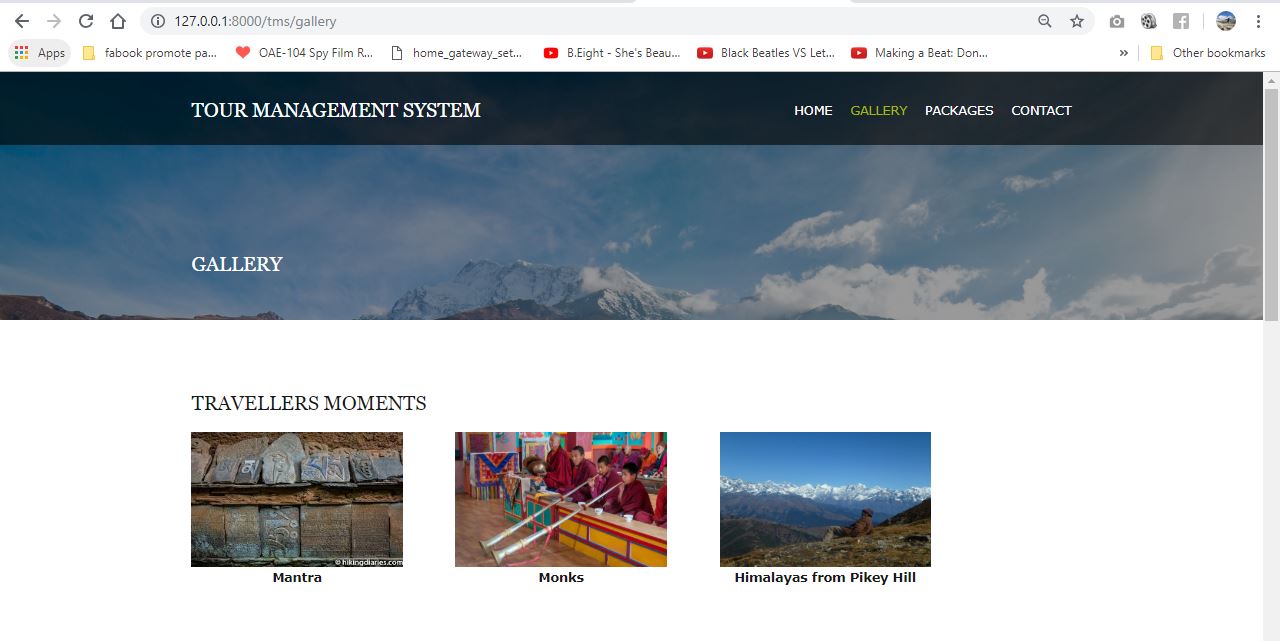


Figure 14Gallery

## **Package**

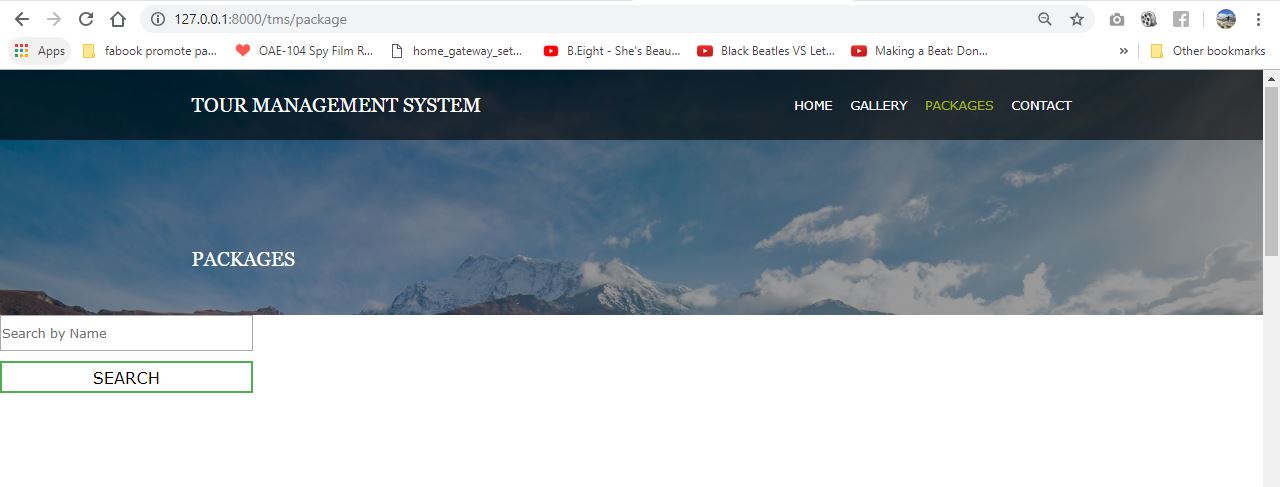


Figure 15Package

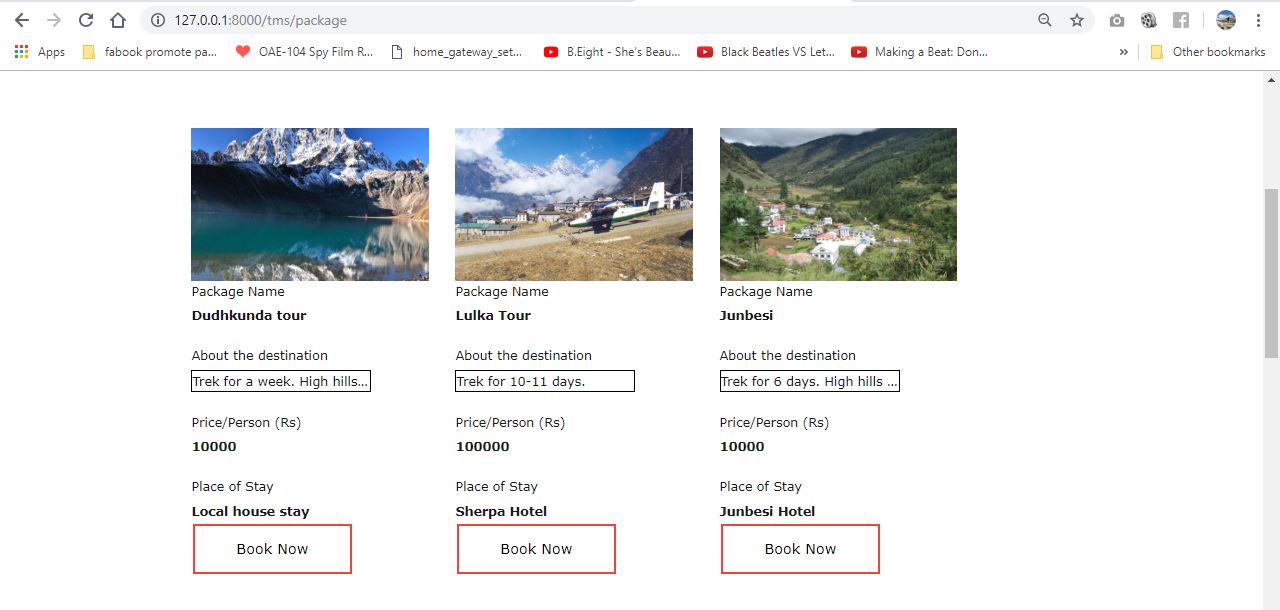


Figure 16Package details

## **Booking**

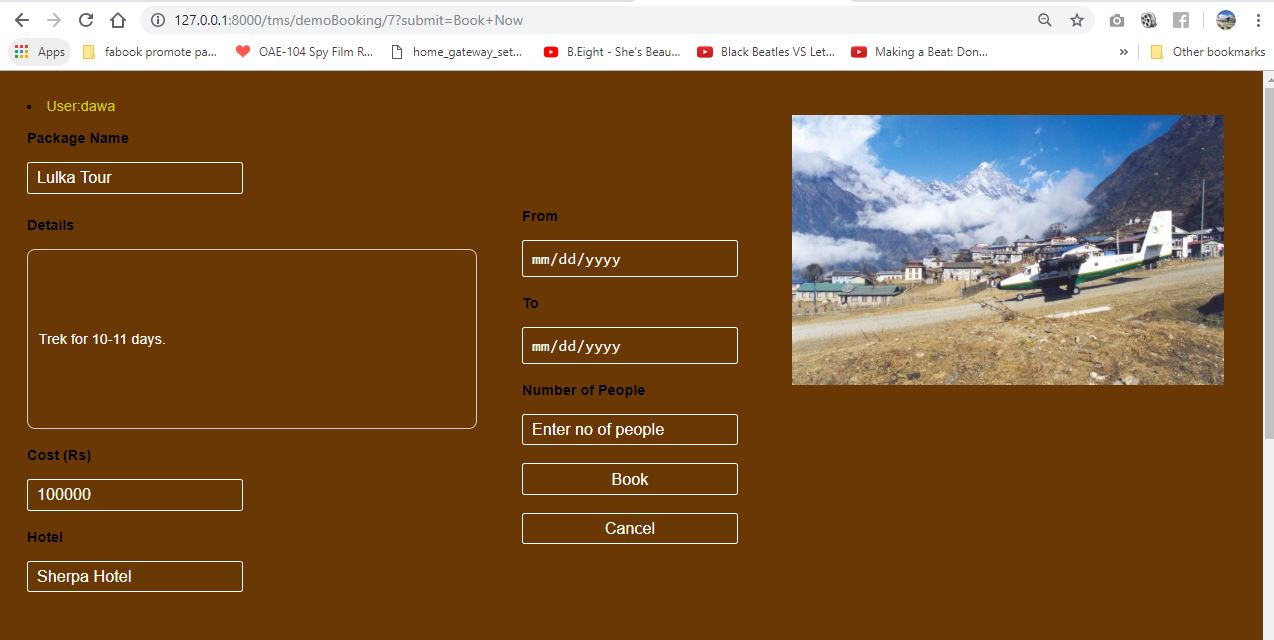


Figure 17Book package

## **Contact**

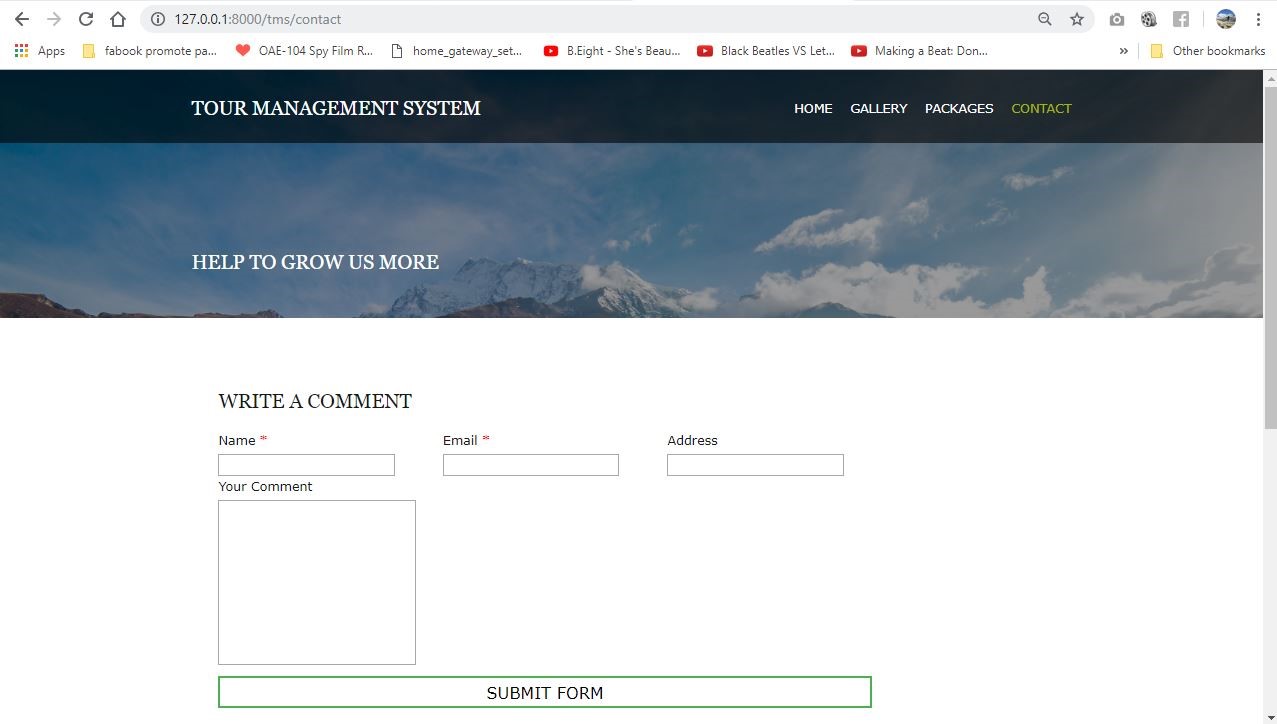


Figure 18Comment box

## **Admin Dashboard**

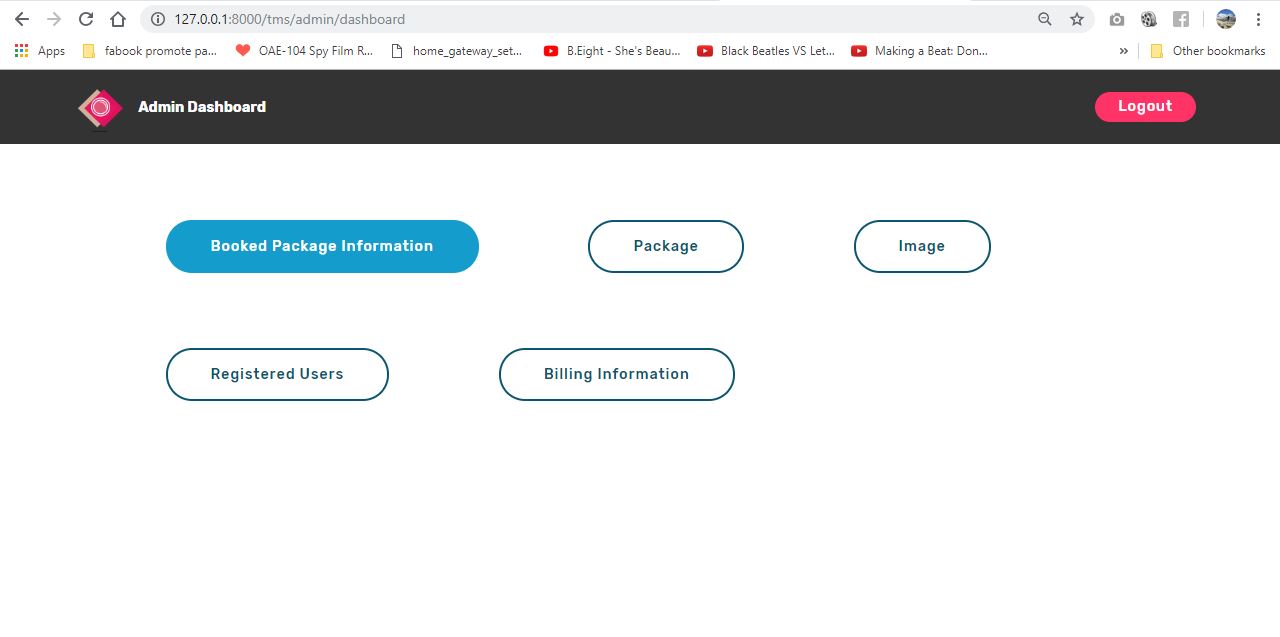


Figure 19Admin page

## **Package Booked Information**

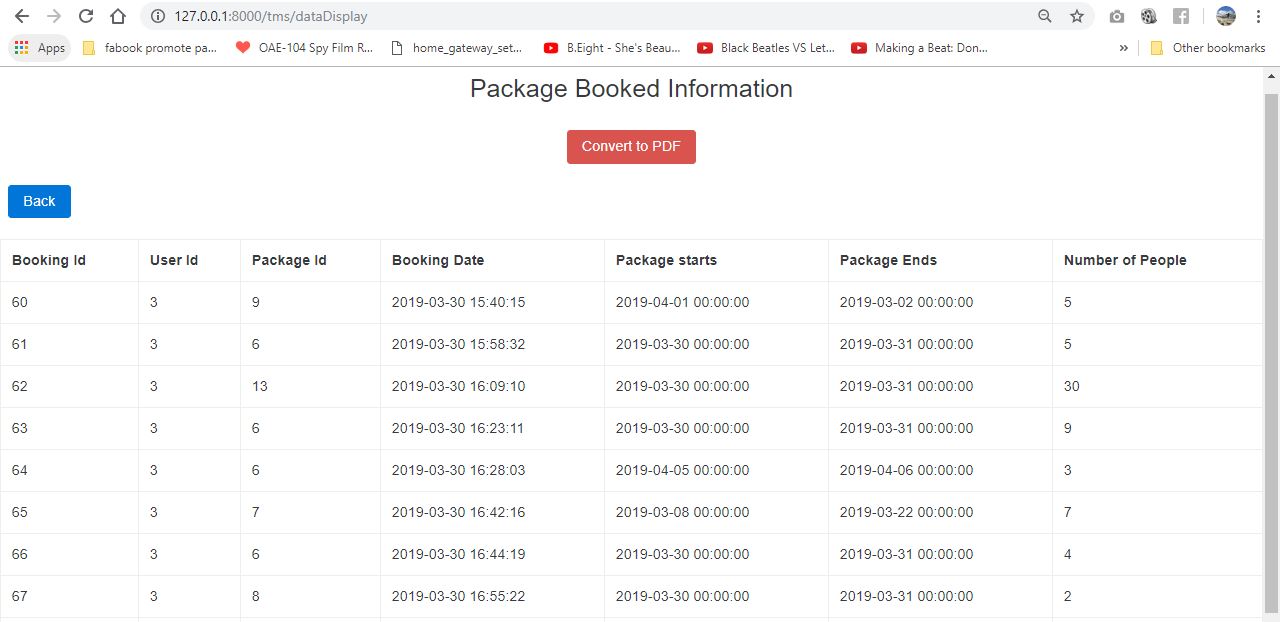


Figure 20Admin package booked information.

## **Admin package CRUD**

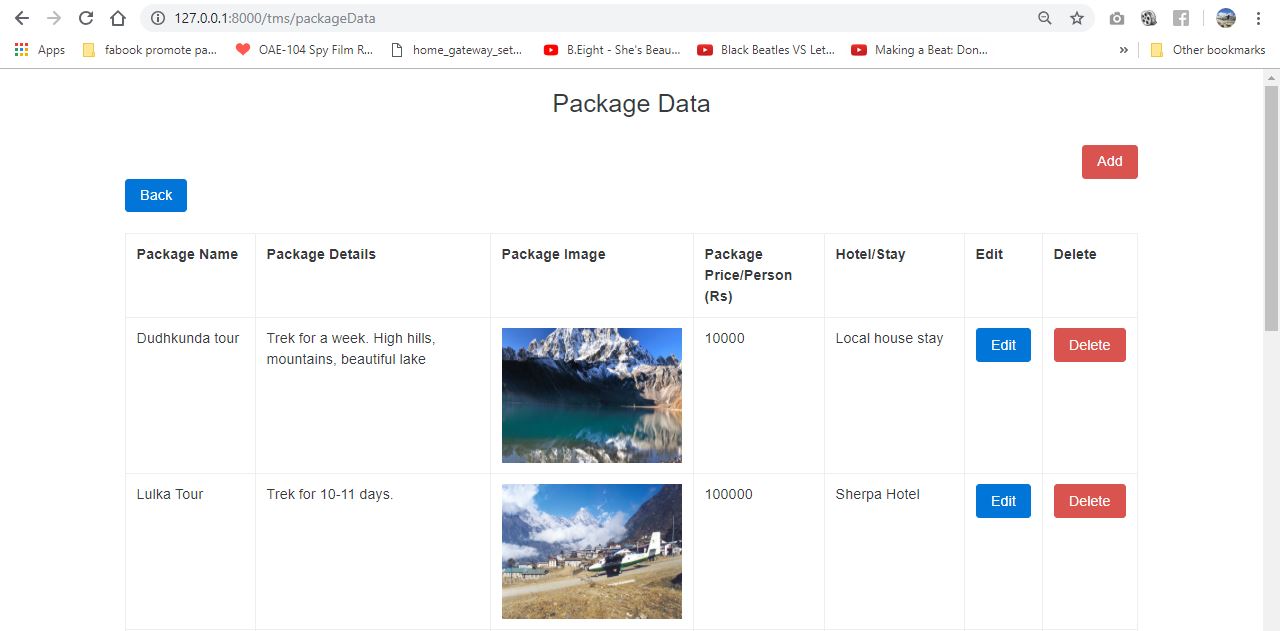


Figure 21Admin Package CRUD

## **Admin Gallery Function**

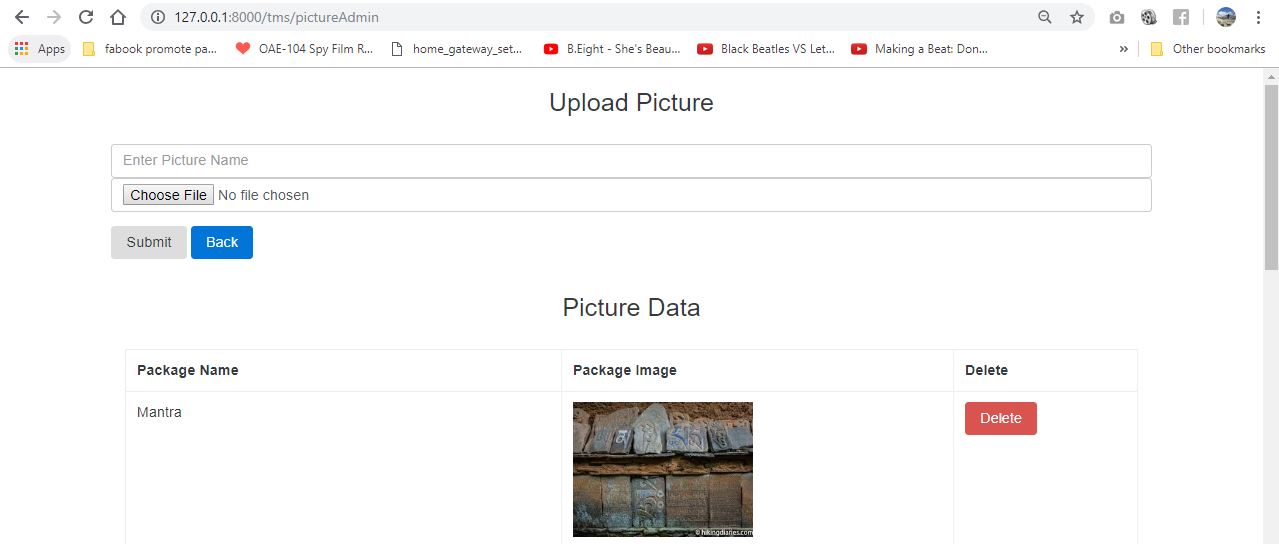
****

Figure 22Admin Gallery Function.

## **User information**

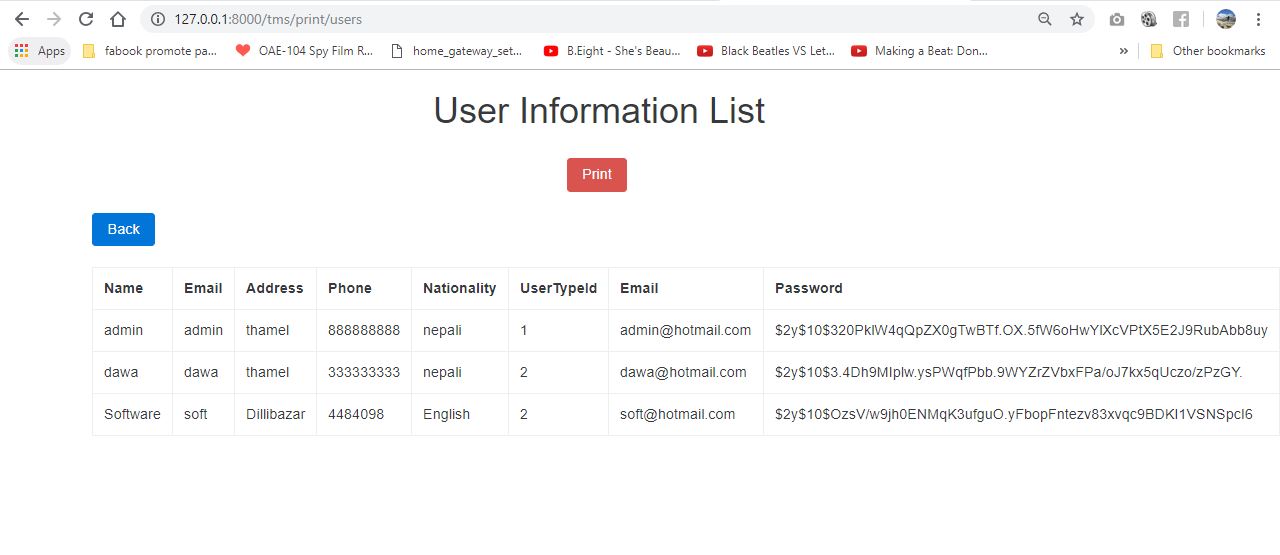


Figure 23User information.

# **Chapter 5 TESTING**

## **Introduction**

Testing is a process of executing a program with the aim of finding error. To make our software perform well it should be error free. If testing is done successfully it will remove all the errors from the software.

## **Principles of Testing**

* All the test should meet the customer requirements.
* To make our software testing should be performed by third party.
* Exhaustive testing is not possible. As we need the optimal amount of testing based on the risk assessment of the application.
* All the test to be conducted should be planned before implementing it.
* It follows pareto rule (80/20 rule) which states that 80% of errors comes from 20% of the program components.
* Starts test with small parts and extend it to large parts.

## **Black box Testing**

It is a [software testing method](http://softwaretestingfundamentals.com/software-testing-methods/) in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.N No.** | **Testcase** | **Expected result** | **Actual result** | **Pass/Fail** | **Action** | **Conclusion** |
| 1 | Login | By entering username and password, user can access the system | System accessed | Pass | No action | Pass |
| 2 | Register | User can redirect to login after filling details on register page | Login form viewed | Pass | No action | Pass |
| 3 | Book Package | User can book the package | Successfully booked | Pass | No action | Pass |
| 4 | Add Package | Package will be added in database | Package added in database | Pass | No action | Pass |
| 5 | Update Package | Package should be updated in database | Package updated in database | Pass | No action | Pass |
| 6 | Delete Package | Package should be deleted from database. | Package deleted from database | Pass | No action | Pass |
| 7 | Add Hotels | Hotel should be added to database | Display error message | Fail | Fix error message | Fail |
| 8 | Logout | User should be logout from the system | Successfully logout | Pass | No action | Pass |
| 9 | Receipt Generate | User should get receipt after booking package | Receipt successfully received | Pass | No action | Pass |
| 10 | Update Profile | User should be able to update profile. | Profile successfully updated | Pass | No action | Pass |

## **Unit Testing**

# **Conclusion**

I have successfully completed the analysis phase of the developing software. In the completion of this phase, we have outcome the requirements, database model and user interface.