CHAPTER 1

INTRODUCTION

1.1 Introduction

The Tours and Travel Management System is a web based application. The main purpose of "Tours and travels management system " is to provide a convenient way for a customer to book tour packages for tour purposes. The objective of this project is to develop a system that automates the processes and activities of a travel agency. In this project, We will make an easier task of searching places and for booking tour packages.

In the present system a customer has to approach various agencies to find details of places and to book tickets. This often requires a lot of time and effort. We provide approach skills to critically examine how a tourist visits and its ability to operate in an appropriate way when dealing with the consequences of tourism, locally, regionally, and nationally including visitor security and ecological influences. It is tedious for a customer to plan a particular journey and have it executed properly.

The project 'Tours and Travels Management System' is developed to replace the currently existing system, which helps in keeping records of the customer details of destination as well as payment received.

This application is develop to provide best tourist services to the customers .We have developed tours and travel management system to provide a search platform where a tourist can find their tour places according to their choices. This system also helps to promote responsible and interesting tourism so that people can enjoy their holidays at their favorable places. This system also helps to develop tourism with different cultures so that they enrich the tourism experience and build pride. We develop this system to create and promote forms of tourism that provide healthy interaction opportunities for tourists and locals and increase better understanding of different cultures, customs, lifestyles, traditional knowledge and believes.

This system also gives tours related information like which places are tourist attractions, cities, and provinces. Tourist can also get the Map and navigation system and temperature and weather information. Tourist can also book tours through our tours and travels management system.

CHAPTER 2

PRODUCT OVERVIEW AND SUMMARY

2.1 Purpose

The objective of the project is to make an application to tourist having provide Online Registration, Tour Package Information, Ticket Booking, Online Payment and Searching Facility for Customer and also Generate Different types of Report. In order to build such an application complete web support need to be provided. A complete and efficient web application which can provide the online booking of tour packages is the basic objective of the project. The web application can be implemented in the form of application with web view. The central concept of the application is to allow the customer to view virtually using the Internet and allow customers to book the packages and tour as per their convenience . The information pertaining to the packages are stores on an mysql at the server side (store).

The application was designed into two modules first is for the customers who wish to view the packages. Second is for the administrator who maintains and updates the information about packages ,booking .The end user of site where the application is hosted on the web and the administrator maintains the database. Once the authorized personnel feed the relevant data into the system, several reports could be generated as per the security.

2.2 Scope

This system can be implemented to tours and travels agency in the locality or to multinational branded company. The system recommends a facility to accept the booking 24*7 which can make customers happy.
If website are providing an online portal where their customers can book easily from anywhere, the website won't be losing any more customers to the trending online tours and travels sites such as kesari or makemytrip.com Since the application is available in the Smartphone it is easily accessible and always available.

2.3 User Classes and Characteristics

Admin

The administrator is the super user of this application. Only admin have access into this admin page. Admin may be the owner of the website. The administrator has all the information about all the users and about all packages , booking .

Characteristics of Admin class:

1. Admin can Add Category.

2. Admin can Add State.

3. Admin can Add Packages.

4.Udpate Category ,State ,Packages.

5.Delete Category, State, Packages.

6.View Enquiry.

7.View Booking Request.

8.Confirm Booking.

9. Admin can Add User.

10. Admin can Update User.

Customer

The customer should be able to perform the following functions using system this

1.Register.
2**.** Login .
3.View Packages.
4.Search Packages.
5.Book the Packages.
6.Payment
7.Send Enquiry

2.4 Design and Implementation Constraints

Design and implementation constraints speak about the restrictions or limitations of design made by the design team. Following are some highlighted points regarding with the design and implementation limitations our project.

- GUI is only in English.
- Login and password is used for the identification of users.
- Only registered customers can book the tour.
- Limited to HTTP/HTTPS protocols.
- This system works only on a single server.

2.4.1 ER Diagram

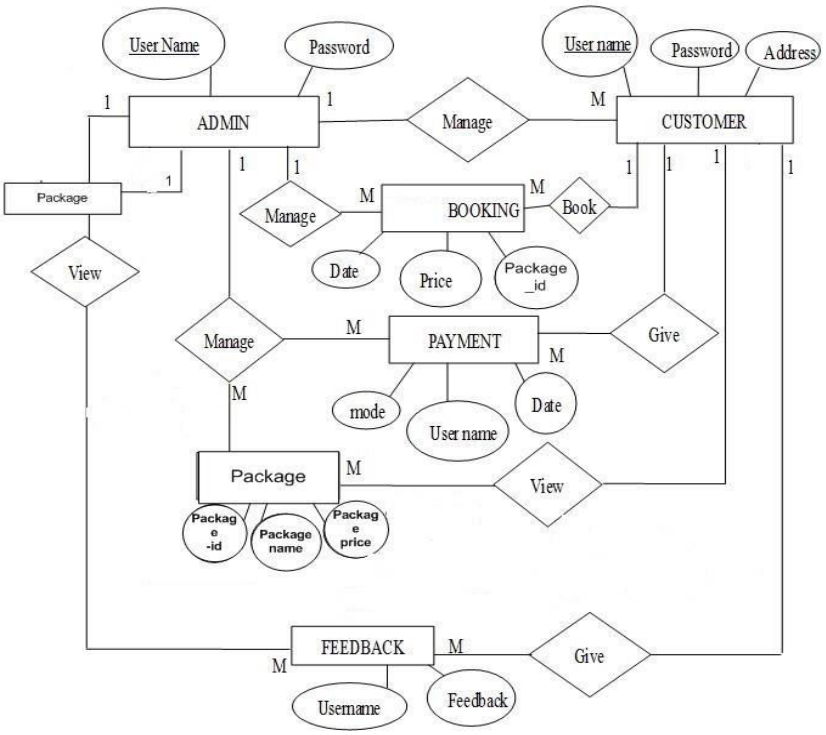


Fig 2.1 ER diagram

2.4.2 Class Diagram

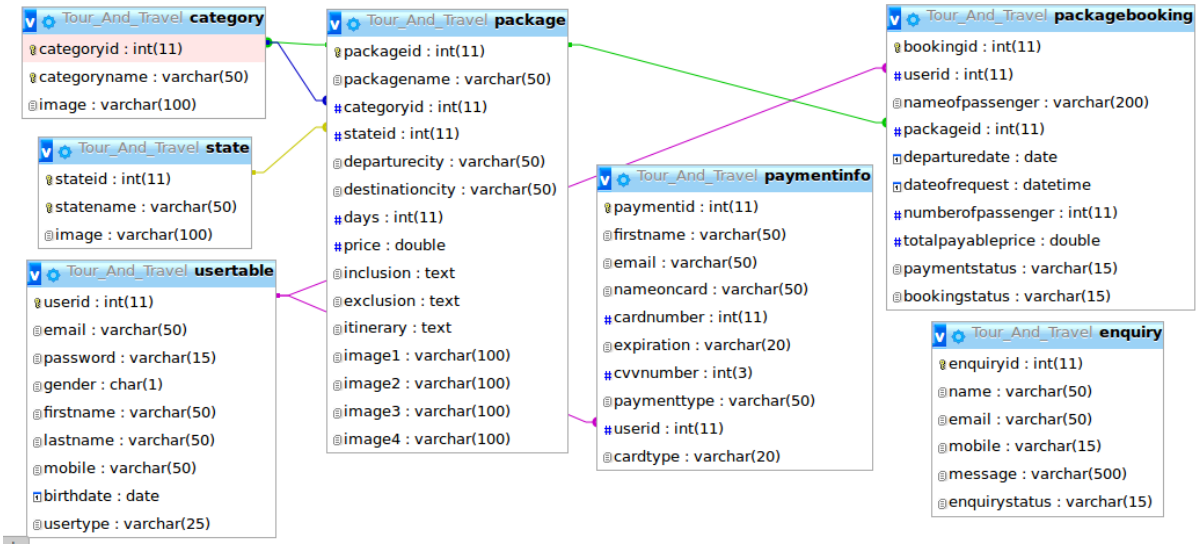


Fig 2.2 Class diagram

CHAPTER 3

REQUIREMENTS

3.1 Functional Requirements

Basically, functional requirements describe the features, functioning, and usage of a product/system/software from the perspective of the product and its user. Although referred to as "requirements," they really are a form of design, high-level. Functional requirements also often are called "functional specifications," and "specification" is a synonym for design.

3.1.1 Use case for Admin

This administrator use case diagram gives us the brief information about the functionalities performed by an Admin.

Briefly following functionalities done by the Administrator as per the bellow diagram.

- Login.
- Registration
- Manage user
- Add, Update, see Details of Packages.
- View Booking Request list
- Approve or dismiss the Booking request.

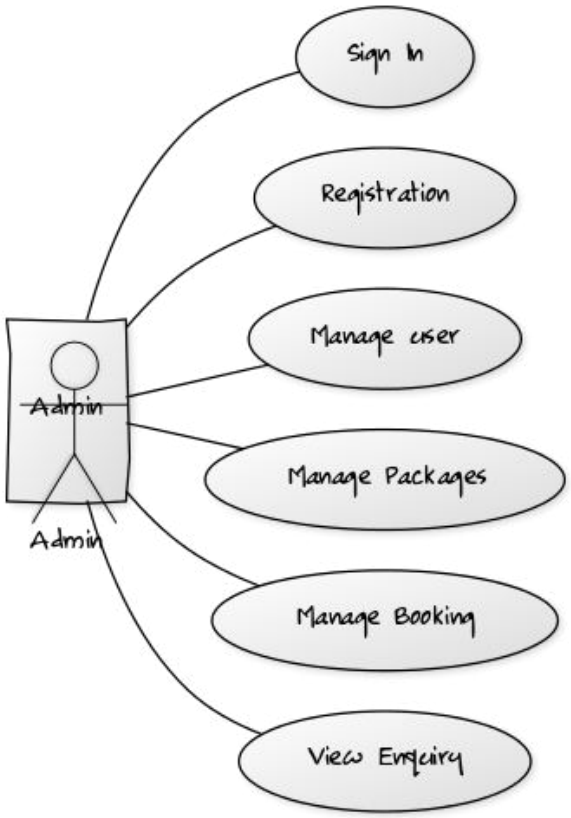


fig3.1.Admin use case

3.1.2 Use case for Customer

This customer use case diagram gives us the brief information about the functionalities performed by an customer.

Briefly following functionalities done by the Administrator as per the bellow diagram.

- Login.
- View Details of Packages.
- Book Package.
- Register.

- Payment.
- Send Enquiry.
- Search Packages.

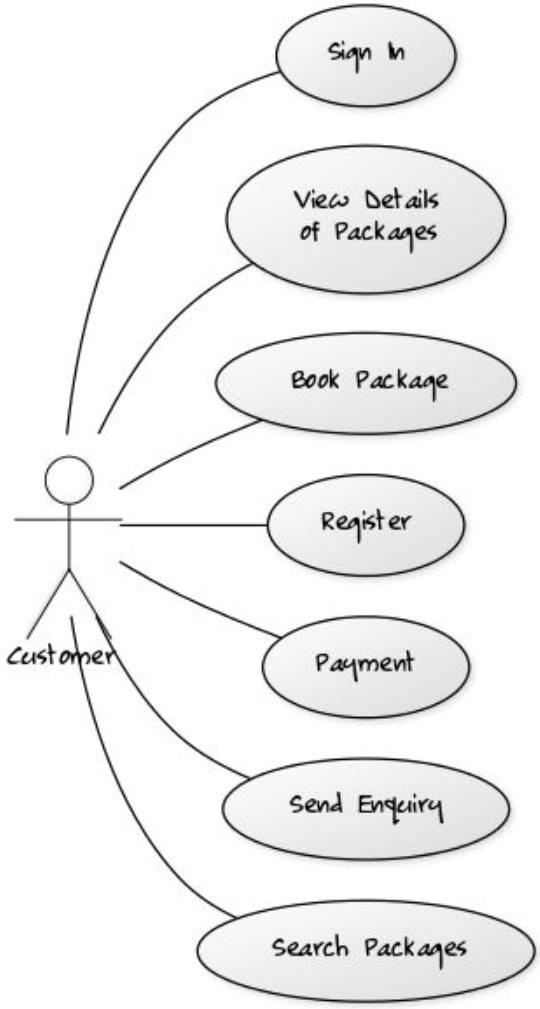


fig 3.2 Use case for customer

3.1.3 Other Requirements

HARDWARE

Processor	**:**	Pentium 2.4 GHz or above
Memory	**:**	256 MB RAM or above
Cache Memory	**:**	128 KB or above
Hard Disk	**:**	10 GB or above [at least 3 MB free space required]

SOFTWARE

Operating System	**:**	Ubuntu 16.04
Font-End Tool	**:**	JSP, Java Script, HTML 5, CSS 3
Back-End	**:**	MySQL

3.2 Non - Functional Requirements

Non-functional requirements are not non-functional at all. Rather, they describe various quality factors, or attributes, which affect the functionality's effectiveness. They do not exist in the abstract but only with respect to relevant functionality. They are often called "ilitis," because many end in "ility," such as, usability, reliability, and maintainability.

For instance, if the software doesn't satisfy relevant usability requirements for applicable functional usage, users can't use it appropriately and thus will not achieve the required functioning. Inadequate usability may cause errors which nullify the value of the functioning, such as miscalculating something. Usability difficulties could cause the user not to use (all) the functions necessary to achieve the value, perhaps because they're not able to employ the necessary functions. It may take so long and be so unpleasant to use the software that the user can't use it as much as is needed or even abandons its use entirely.

3.2.1 Usability Requirement:

Usability is the ease of use and learn ability of a human-made object. The object of use can be a software-application, website, book, tool, MACHINE, process, or anything a human interacts with. A usability study may be conducted as a primary job function by a usability analyst or as a secondary job function by designers, technical writers, marketing personnel, and others. It is widely used in consumer electronics, communication, and knowledge transfer objects and mechanical objects such as a door handle or a hammer.

Usability includes methods of measuring usability, such as needs analyst and the study of the principles behind an object's perceived efficiency or elegance. In human-computer interaction and computer science, usability studies the elegance and clarity with which the interaction with a computer program or a web site is designed. Usability differs from user satisfaction and user experience because usability also considers usefulness.

Usability Requirements for this project design should support the following from the perspective of its primary users:

- Efficiency of use: goals are easy to accomplish quickly and with few or no user errors
- Intuitiveness: the interface is easy to learn and navigate; buttons, headings, and help/error messages are simple to understand
- Low perceived workload: the interface appears easy to use, rather than intimidating, demanding and frustrating.

3.2.2 Performance Requirement:

Performance requirement within system engineering, encompasses the set of roles, skills, activities, practices, tools, and deliverables applied at every phase of the systems development life cycle which ensures that a solution will be designed, implemented, and operationally supported to meet the non-functional requirements for performance.

In our project basic performance requirements get maintain by using object oriented concepts such as major pillars of object oriented programming thought process are ‘Abstraction’, ‘Modularity’, ‘Encapsulation’, ‘Hierarchy’ and minor pillars are ‘Concurrency’, ‘Persistence’, ‘Typing’. These pillars keep the inter-functionalities for the project as ‘loosely coupled and highly cohesive’.

3.2.3 Reliability Requirement:

Reliability Requirement is requirement that emphasizes dependency in the lifecycle management of a product. Dependability, or reliability, describes the ability of a system or component to function under stated conditions for a specified period of time.Reliability requirement may also describe the ability to function at a specified moment or interval of time (Availability). Reliability is defined as the probability of success (Reliability=1-Probability of Failure), as the frequency of failures; or in terms of availability, as a probability derived from reliability, testability and maintainability. Testability, Maintainability and maintenance are often defined as a part of "reliability engineering" in Reliability Programs. Reliability plays a key role in the cost-effectiveness of systems.

In this project, to achieve the reliability requirement of the software the various programming layers have to reliable on each other inclusively.

For example in our project, we have database, POJO layer, DAO layer, Controller layer, View layer. Among these, each layer has to relay on other layer. For example, POJO layer has dependency on database, for the implementation of DAO layer it has to take some help of POJO layer and so on. This activity shows the reliability requirements of our project.

3.2.4 Portability Requirement:

Portability in high-level computer programming is the usability of the same software in different environments. The pre-requirement for portability is the generalized abstraction between the application logic and system interfaces. When software with the same functionality is produced for several computing platforms, portability is the key issue for development cost reduction. In our project, we are making it using Java technology i.e. using spring, hibernate MVC etc. if user who is going to use this project application has different operating environment then it won't matter unless and until user has compatible JVM(Java Virtual Machine) for his/her operating system. Hence through this we achieved the Portability Requirement.

3.2.5 Security Techniques:

The term security requirement is used by different communities and groups in different ways and may require additional explanation to establish the particular context for the various use cases. Security requirements can be stated at a very high level of abstraction, FISMA and FIPS 200 articulate security requirements at such a level. Organizations take these high-level security requirements and define certain security capabilities needed to satisfy those requirements and provide appropriate mission/business protection.

Security requirements are also reflected in various nontechnical security controls that address such matters as policy and procedures at the management and operational elements within organizations, again at differing levels of detail. It is important to define the context for each use of the term security requirement so the respective communities (including individuals responsible for policy, architecture, acquisition, engineering, and mission/business protection) can clearly communicate their intent.

In our project we are not going to use the security facilities explicitly. To provide the security requirement explicitly we have to use techniques such as cryptography and tamper-resistant hardware that can be used to build trust in software tools and processes.Theseopportunities arise from the fact that software systems are no longer monolithic single-vendor creations. Increasingly, systems are complex, late-bound assemblages made up of commercial, off-the-shelf (COTS) elements and even mobile code. COTS offers great savings over custom-written software.
In our project, we used Java platform, Java technology/platform provides security of code up to some extents. Java is secured because of following:

- No explicit pointer.
- Programs run inside virtual machine sandbox.

- Class loader- adds security by separating the package for the classes of the local file system from those that are imported from network sources.
- Bytecode Verifier- checks the code fragments for illegal code that can violate access right to objects.
- Security Manager- determines what resources a class can access such as reading and writing to the local disk.

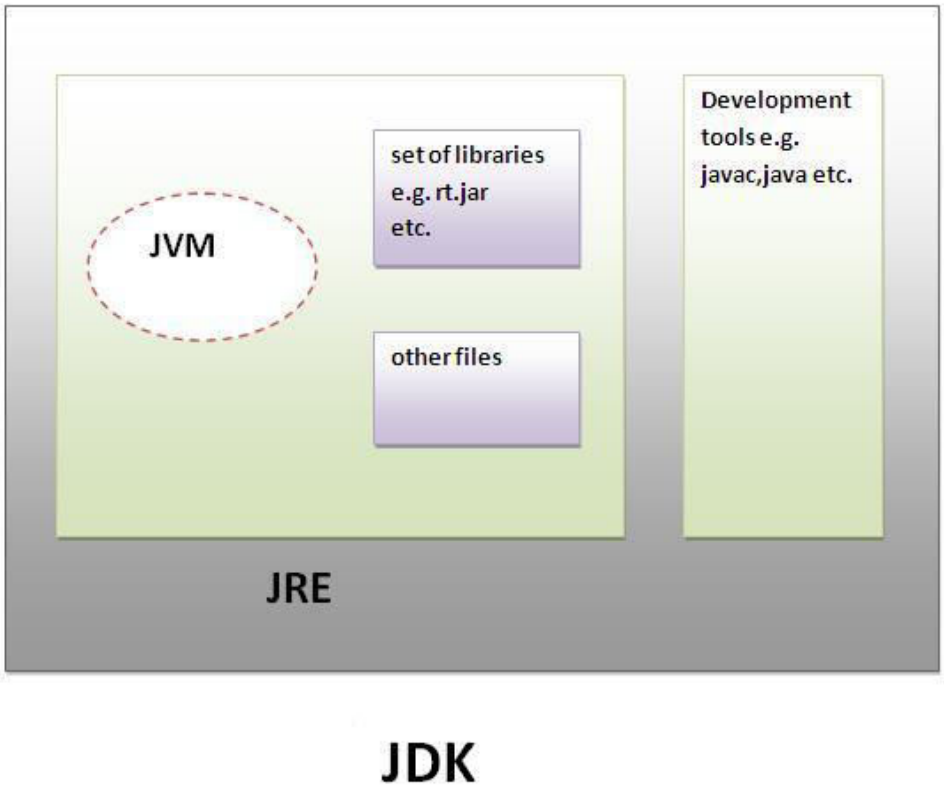


fig3.3Security Environment in Java

CHAPTER 4

PROJECT DESIGN

4.1 Data Model

A data model organizes data elements and standardizes how the data elements relate to one another. Since data elements document real life people, places and things and the events between them, the data model represents reality. A data model is a set of symbols and text used for communicating a precise representation of an information system. A data model provides the details of information to be stored, and is of primary use when the final product is the generation of computer software code for an application or the preparation of a functional specification to aid a computer software make-or-buy decision. The figure is an example of the interaction between process and data models.

Data models are often used as an aid to communication between the business people defining the requirements for a computer system and the technical people defining the design in response to those requirements. They are used to show the data needed and created by business processes.

The data model simply describes the structure of data entities and their relationships. For example,in a banking system, entities will typically include Account, Customer and Loan. Account has several attributes, such as account number, type (savings or checking), status, and current balance.A relationship may dictate that one customer can have one or more accounts, and one account is associated to one or two customers.Data modelling spans the evolution of the high-level model that displays the data entities in a given business domain into a model that shows details of how the data is stored. There are different stages of data models:

- Conceptual data model.
- Logical data model.
- Physical data model.

4.1.1 Database Design

Database design is the process of producing a detailed data model of a database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the

tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system (DBMS).

The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Usually, the designer must:

- Determine the data to be stored in the database.
- Determine the relationships between the different data elements.
- Superimpose a logical structure upon the data on the basis of these relationships.

Within the relational model the final step above can generally be broken down into two further steps that of determining the grouping of information within the system, generally determining what are the basic objects about which information is being stored, and then determining the relationships between these groups of information, or objects. This step is not necessary with an object database.

Table name:usertable

Sr No	Column Name	Data Type	Description
1	userid	int(11)	Primary key
2	firstname	varchar(50)	
3	lastname	varchar(50)	
4	email	varchar(200)	
5	password	varchar(50)	
6	gender	varchar(1)	
7	mobile	int(15)	
8	birthdate	Date	
9	usertype	varchar (15)	

Table 4.1 UserDetails

Table name:Category

Sr No	Column Name	Data Type	Description
1	categoryId	int(11)	Primary key
2	categoryName	varchar(50)	

Table 4.2 Category

Table name: State

Sr No	Column Name	Data Type	Description
1	stateId	int(11)	Primary key
2	stateName	varchar(50)	

Table 4.3 State

Table name: enquirytable

Sr No	Column Name	Data Type	Description
1	enquiryid	int(11)	Primary key
2	name	varchar(50)	
3	email	varchar(200)	
4	mobile	int(15)	
5	message	varchar(500)	
6	enquirystatus	varchar(15)	

Table 4.4 enquiry

Table name: packagetable

Sr No	Column Name	Data Type	Description
1	packageid	int(11)	Primary key
2	packagename	varchar(50)	
3	categoryid	int(11)	Foreign key
4	stateid	int(11)	Foreign key
5	departurecity	varchar(50)	
6	destinationcity	varchar(50)	
7	days	int(11)	
8	price	Float(7,2)	
9	exclusion	varchar(2000)	
10	inclusion	varchar(2000)	
11	itinerary	varchar(2000)	
12	image1	varchar(50)	
13	image2	varchar(50)	
14	image3	varchar(50)	
15	image4	varchar(50)	

Table 4.5 packageDetail

Table name:bookingtable

Sr No	Column Name	DataType	Description
1	bookingid	int(11)	Primary key
2	userid	int(11)	Foreign key
3	nameofpassenger	varchar(50)	
4	packageid	int(11)	Foreign key
5	departuredate	date	
6	dateofrequest	datetime	
7	numberofpassenger	int(11)	
8	totalpayableprice	Float(7,2)	
9	paymentstatus	varchar(50)	
10	bookingstatus	varchar(50)	

Table 4.6 booking

Table name:paymentinfo

Sr No	Column Name	Data Type	Description
1	paymentid	int(11)	Primary key
2	firstname	varchar(50)	
3	email	varchar(200)	
4	nameoncard	varchar(50)	

5	cardnumber	int(11)	Foreign key
6	expiration	varchar(50)	
7	cvvnumber	int(3)	
8	paymenttype	varchar(50)	
9	userid	int(11)	
10	cardtype	varchar(50)	

Table 4.7 Payment

4.1 Functional Decomposition diagram

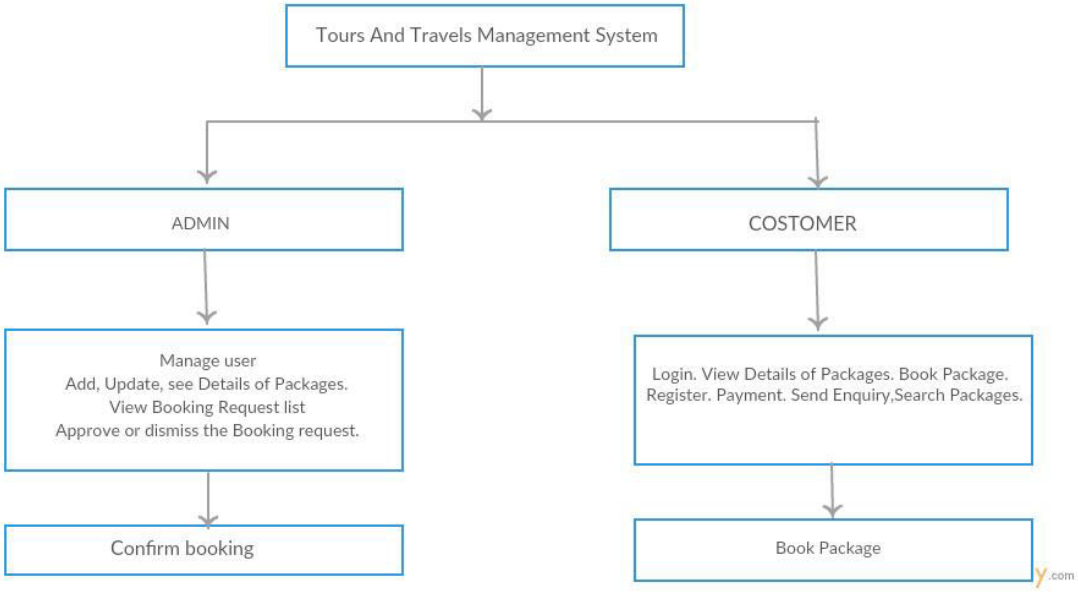


Fig4.1Functional Decomposition diagram

4.2 Data Flow Diagram

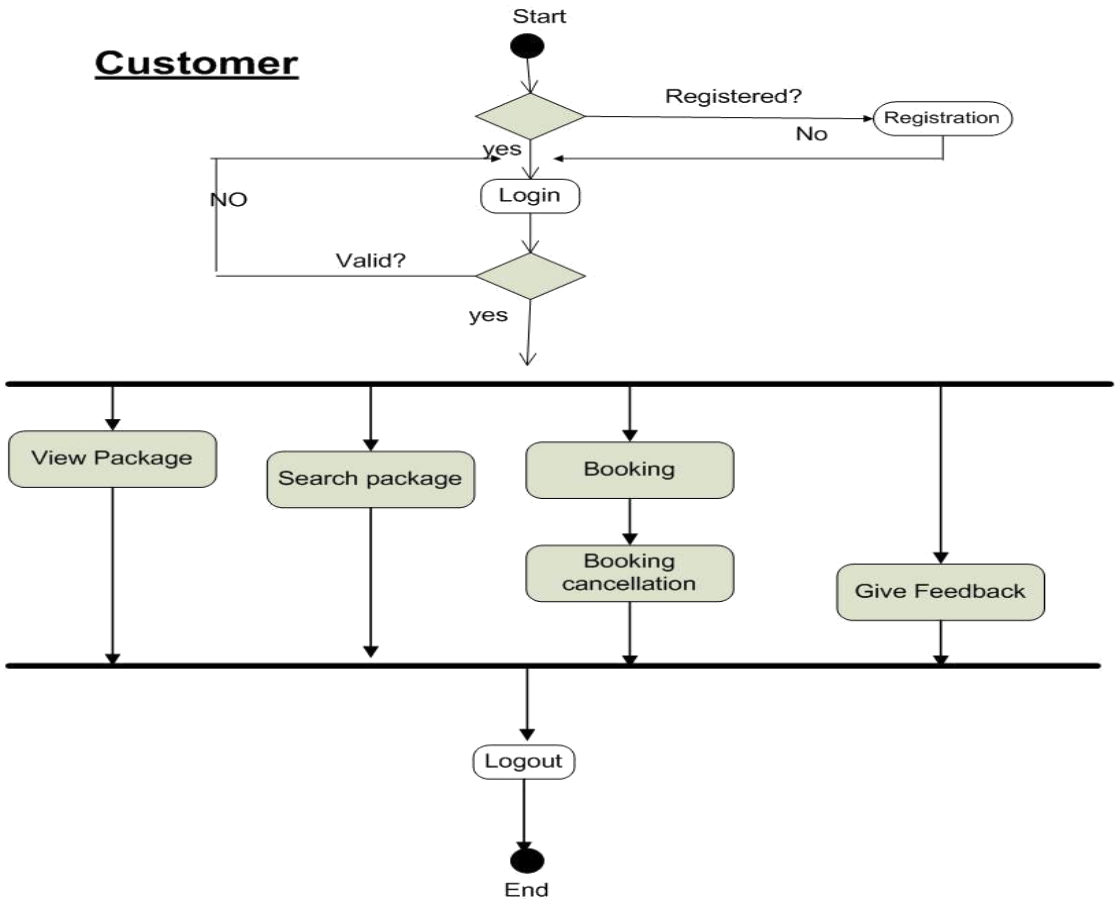


Fig 4.2 customer DFD

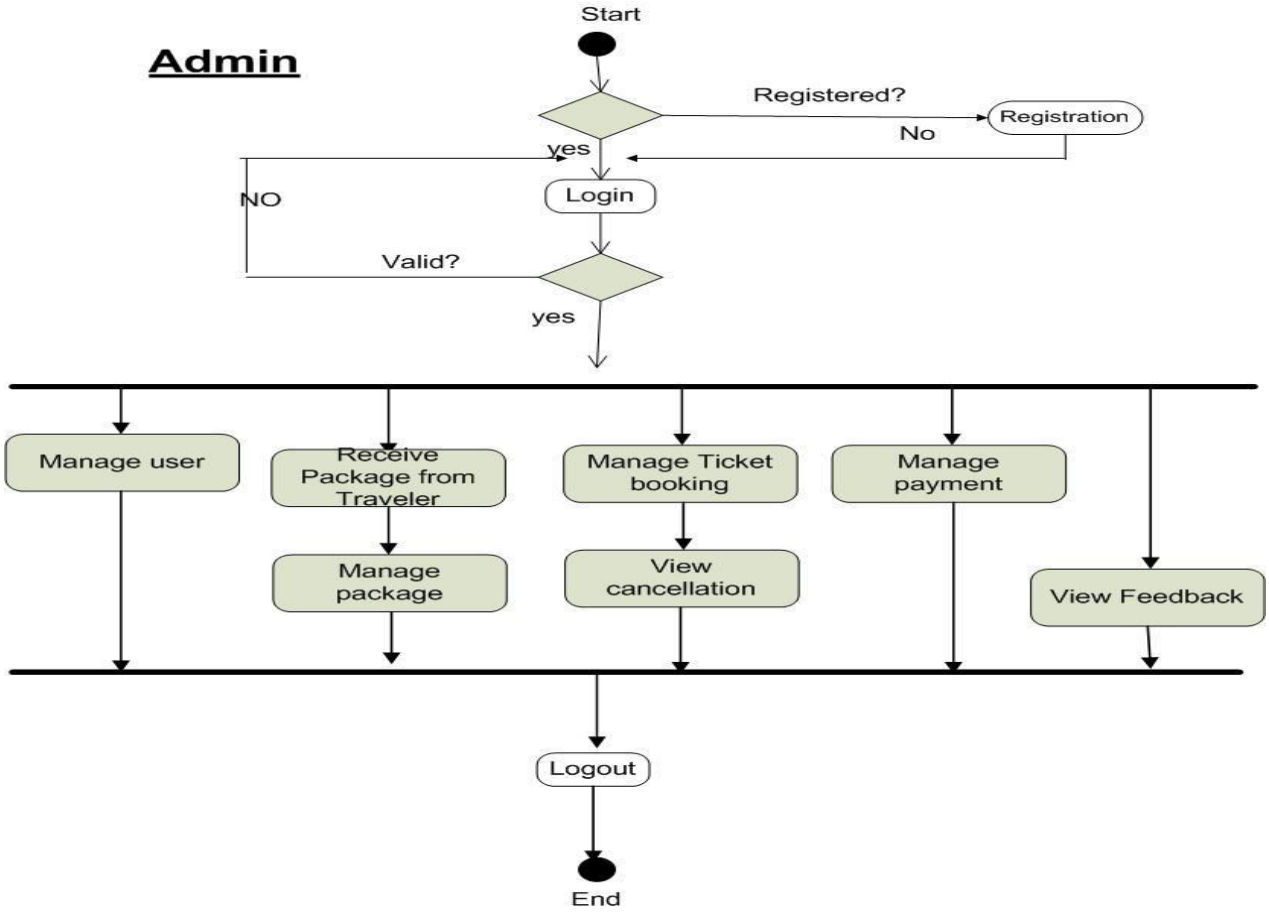


Fig 4.3Admin DFD

4.4 Sequence Diagram

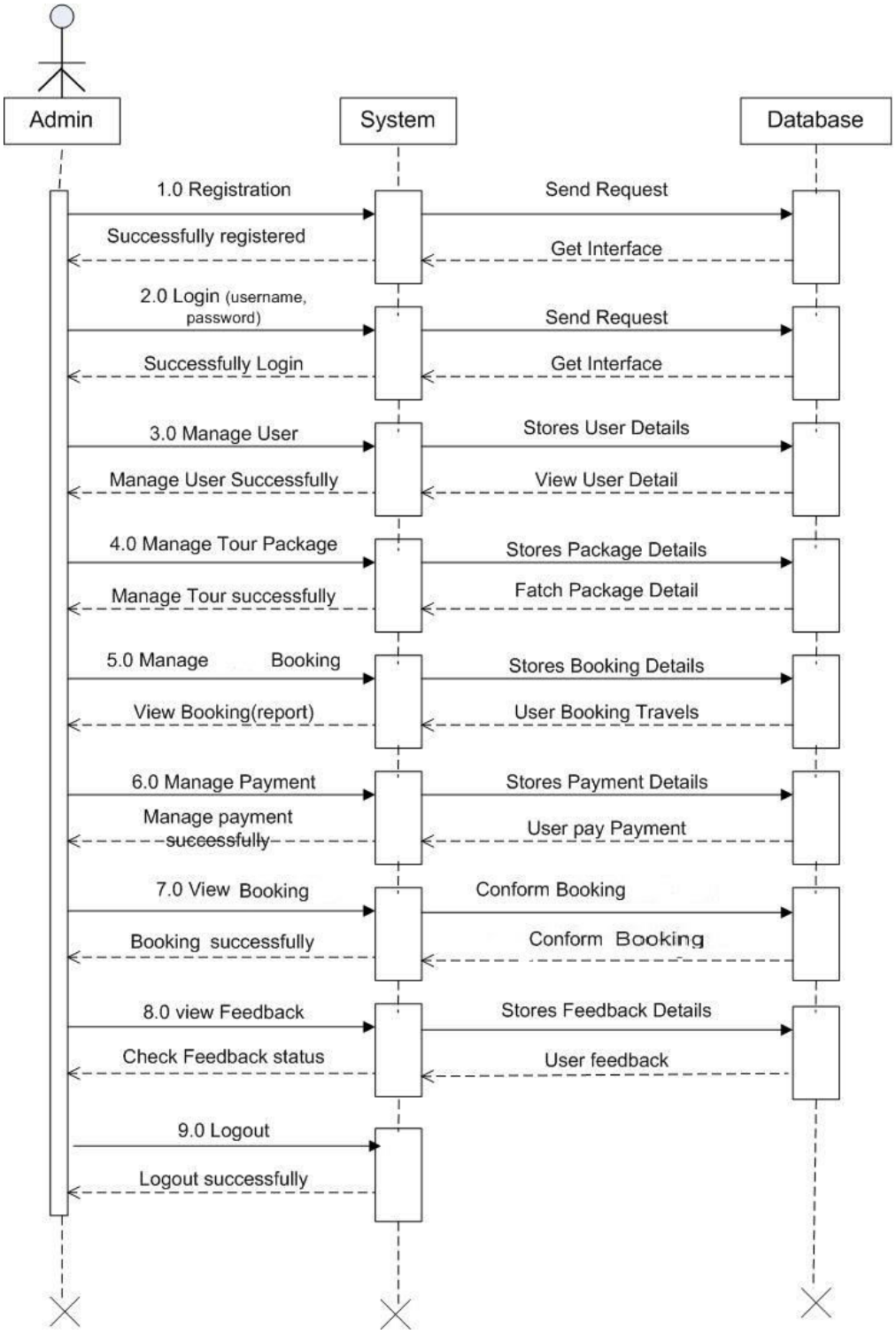


Fig 4.4 Sequence Diagram

CHAPTER 5

TEST REPORT

SR No	Test Case Title	Description	Expected Output	Error Message	Result
1	Login page-Customer	If User Id=User Id Password=Customer Password	If validate allow for Home Page	Invalid Login Please try again	Passed
2	Login page-Admin	If user id = admin And password=admin	If validate allow for Admin Home Page	Invalid Admin And redirected to the same Page	Passed
3	Home Page-Admin	Home page displayed all crud Operation and approve req.	Home page Displayed	No error	Passed
4	New Customer Registration	Is not allowed to keep any field empty.	If validate go to login page	Validation Error	Passed
5	Forget Password-Customer	Empty password not Allowed	Updated in the database	No error	Passed
6	Customer Home page	Show all products And add to cart, buy option	Page displayed	No error	Passed

Table 5.1 TEST REPORT

Chapter 6

PROJECT RELATED STATISTICS

DATE	WORK PERFORMED	SLC Phase	Additional Notes
Dec 28 2016	Project Allotment and User Requirements Gathering	Feasibility Study	----------
Dec 29 2016	Initial SRS Document Validation And Team Structure Decided	Requirement Analysis (Elicitation)	The initial SRS was presented to the client to understand his requirements better
Dec 30 2016	Designing the use-cases, Class Diagram, Collaboration Diagram, E-R Diagram and User Interfaces	Requirement Analysis & Design Phase	Database Design completed
Dec 31 2016	Business Logic Component design Started	Design Phase	----------
Jan 16 2017	Coding Phase Started	Coding Phase	Class Library implemented.
Jan 16 2017	Implementation of Web Application	Coding Phase	Class Library Development going on.
Jan 16,17,18 2017	Implementation of Web Application	Coding Phase	Class Library Development going on.

Jan 19 2017	Implementation of Web Application and Window Application Continued	Coding Phase and Unit Testing	Class Library Modified as per the need.
Jan 19 2017	Implementation of Web Application and Window Application Continued	Coding Phase and Unit Testing	--
Jan 20 2017	After Ensuring Proper Functioning the Required Validations were Implemented	Coding Phase and Unit Testing	Module Integration was done by the Project Manager
Jan 21 2017	The Project was Tested by the respective Team Leaders and the Project Manager	Testing Phase (Module Testing)	--
Jan 22 2017	The Project was Submitted to Other Project Leader of Other Project Group For Testing	Testing Phase (Acceptance Testing)	The Project of Other Team was Taken up by the Team for Testing
Jan 23, 2017	The Errors Found were Removed	Debugging	The Project was complete for submission
Jan 28, 2017	Final Submission of Project		

Table 6.1 PROJECT RELATED STATISTICS

CHAPTER 7

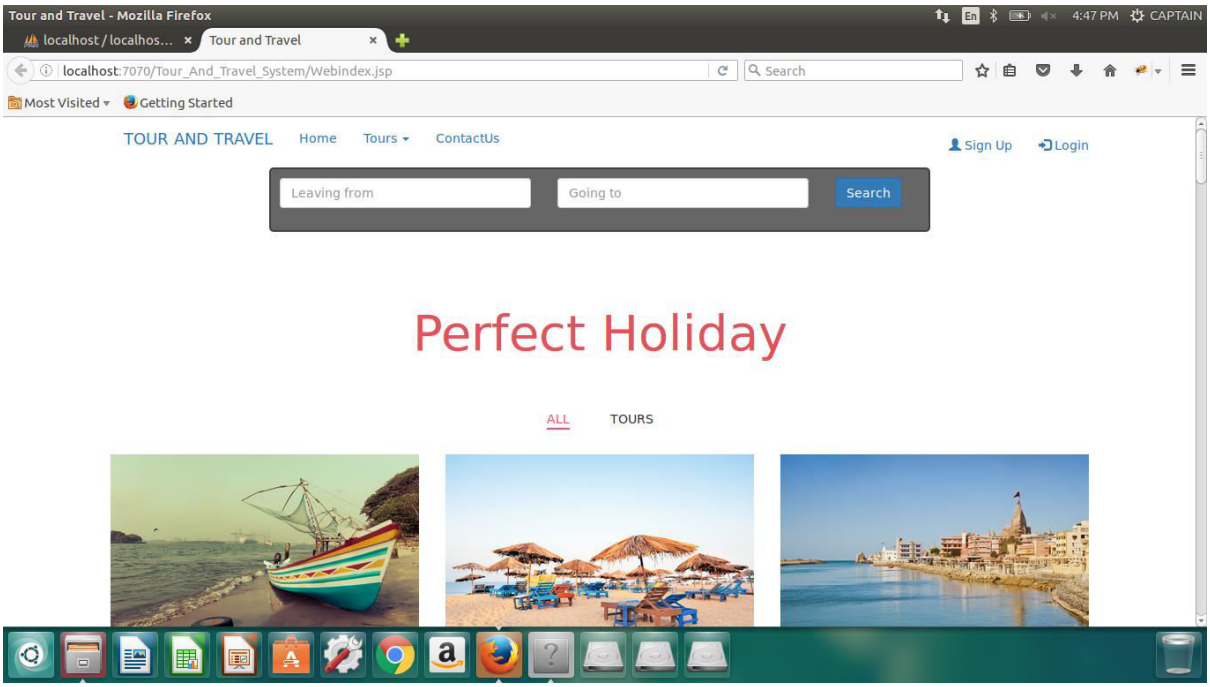
CONCLUSION

The Main objective of the Project is to design and develop a website for Tours and travels . It helps user to Tour packages info details available in the website. It provides security by giving different permissions to the user using the website and thus providing security among the users. It also gives right to any customer to contact us as well as register them- selves with our website to have rights to update details of their profile added by them.

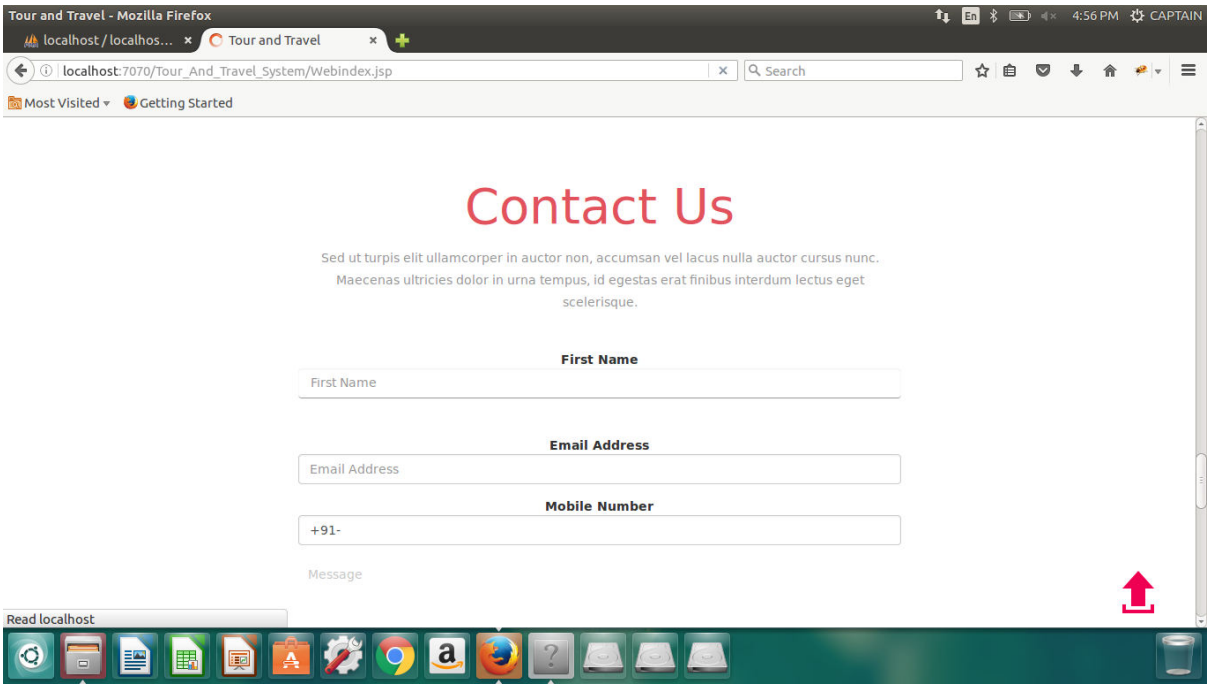
It also provides a way by which any customer can made request to book a tour.

Appendix

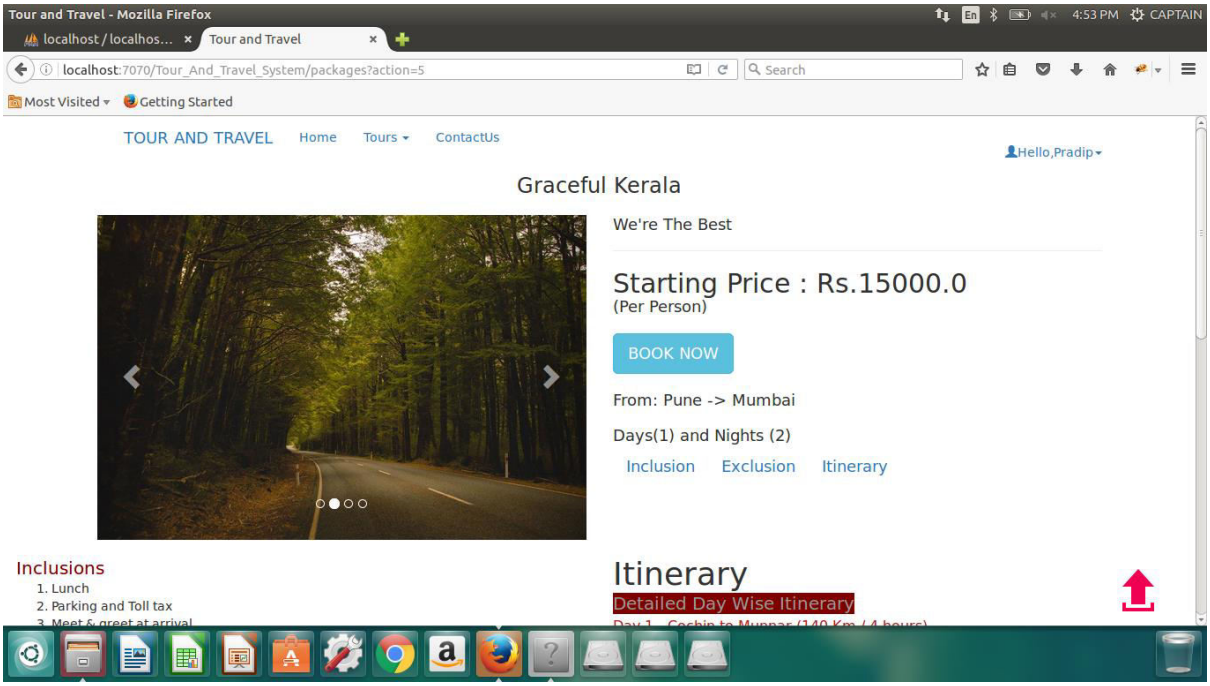
Home Page:



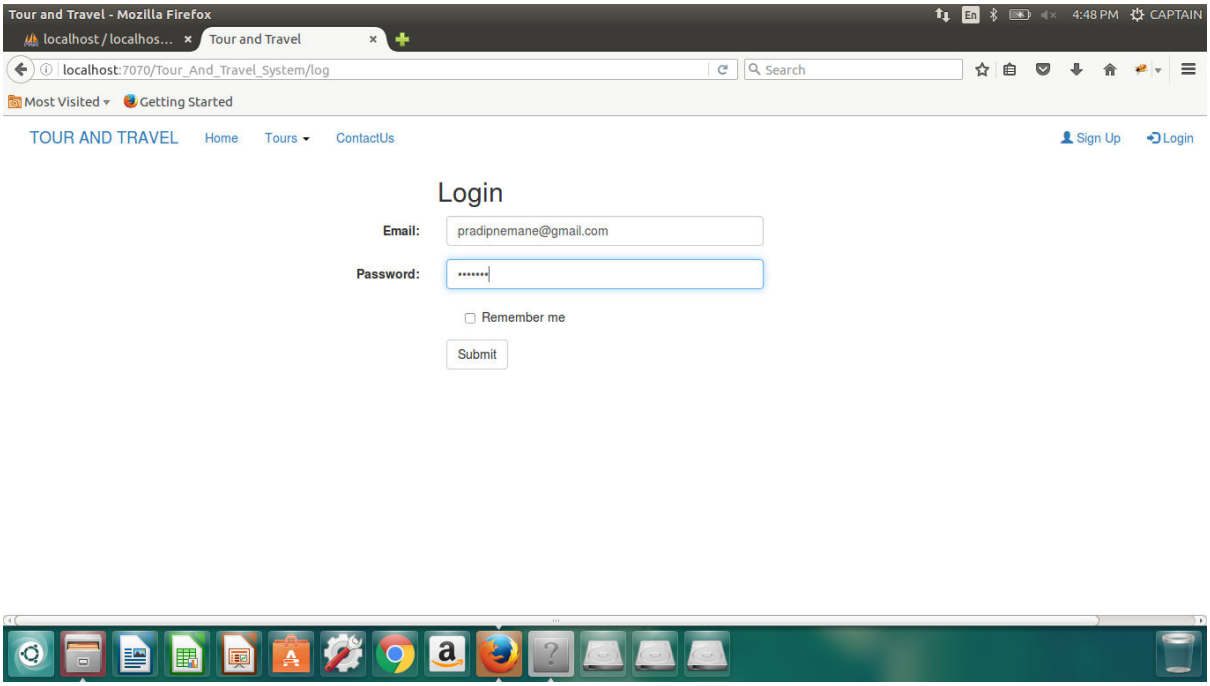
ContactUs:



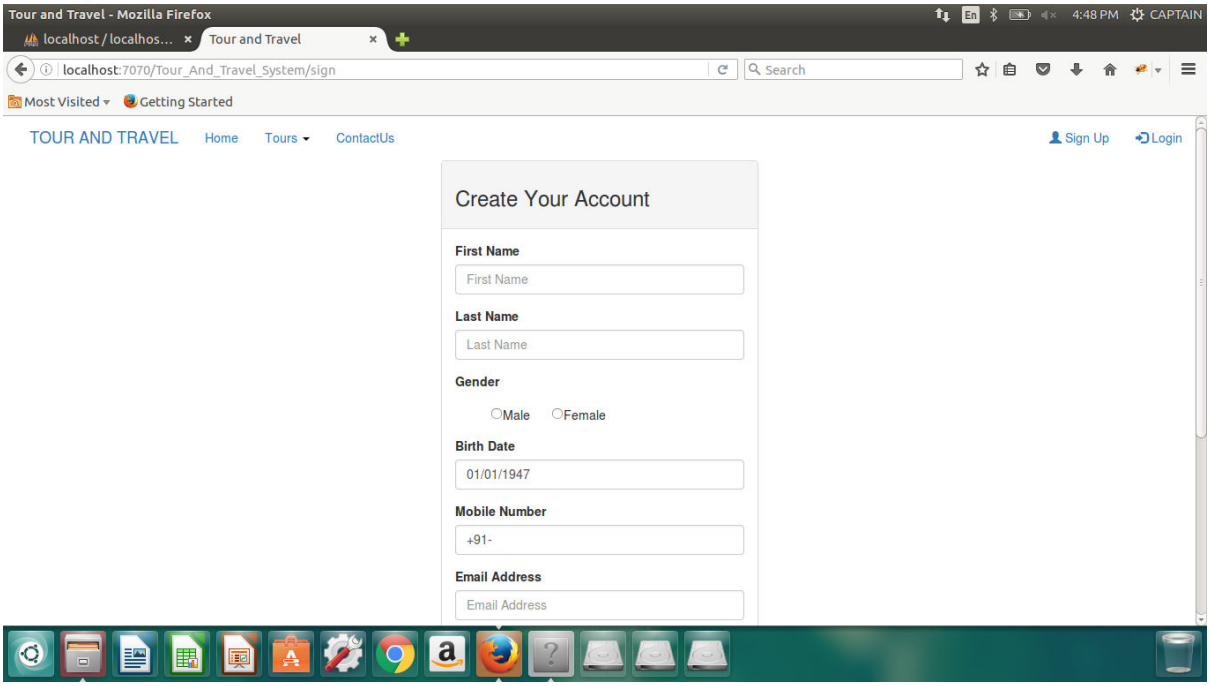
Packageinfo:



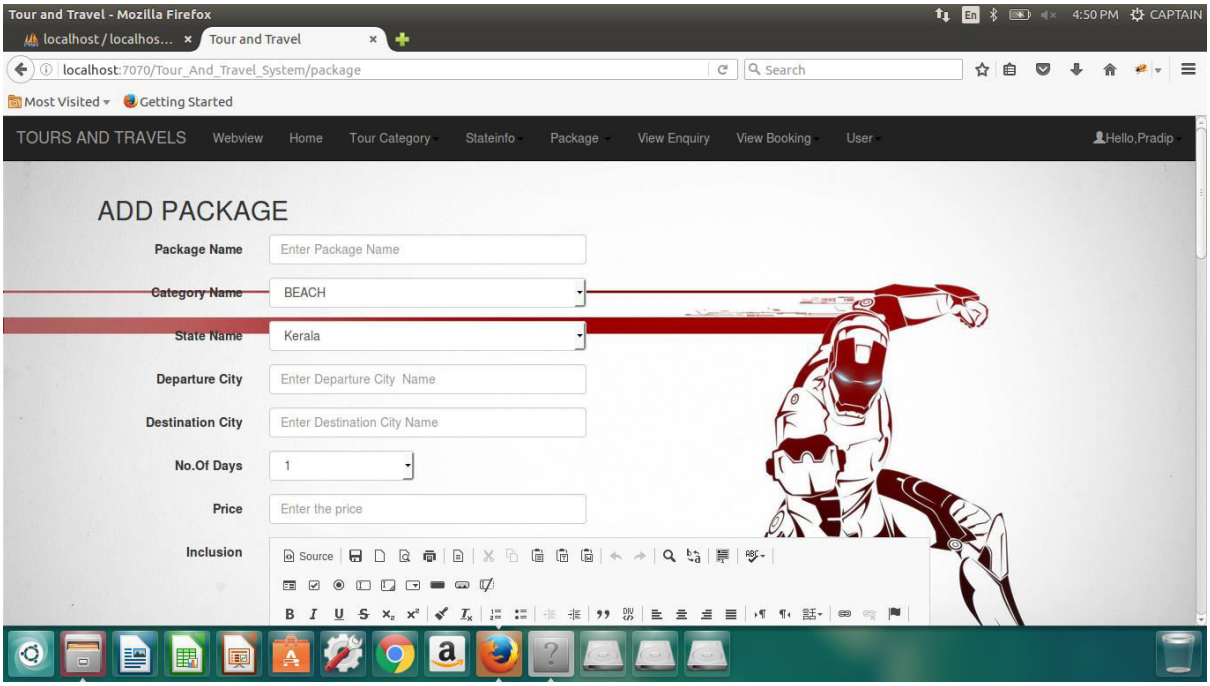
Admin Login:



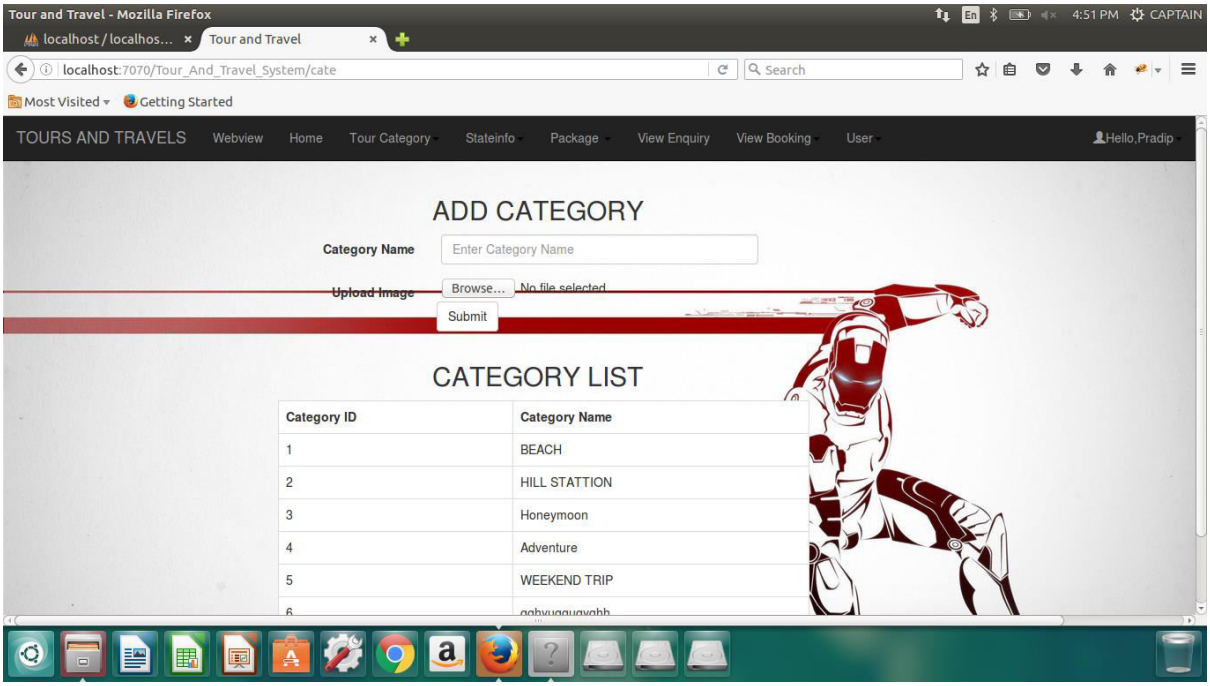
Registration:



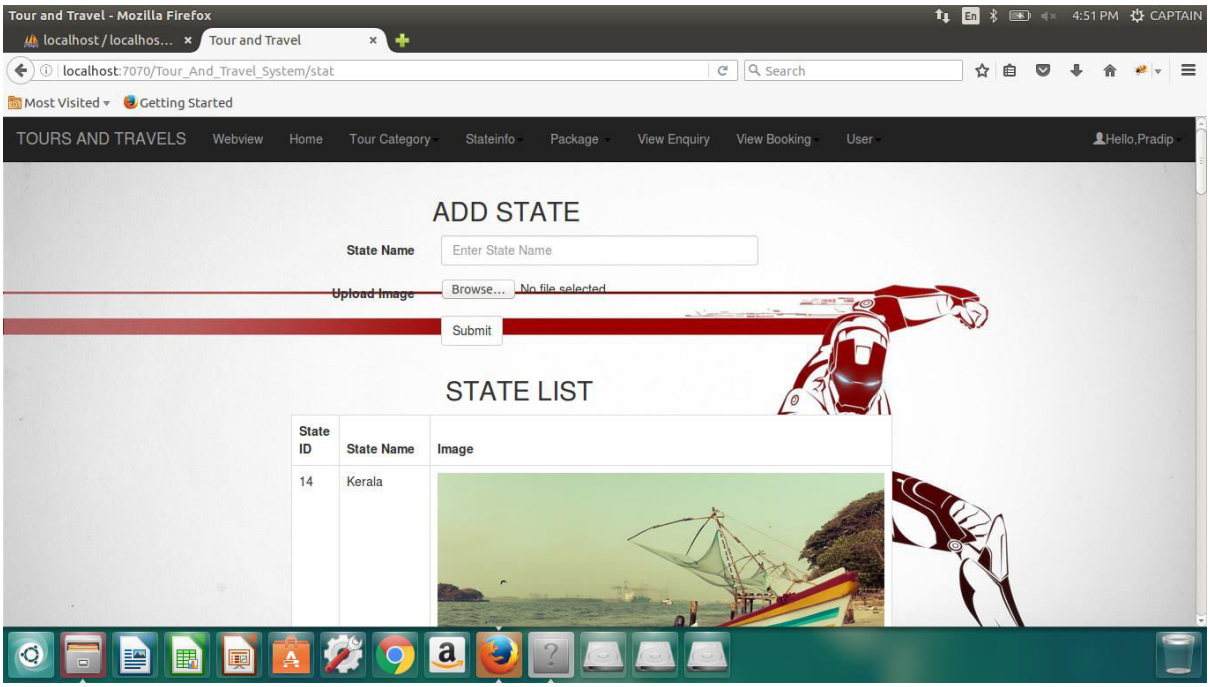
Add Package:



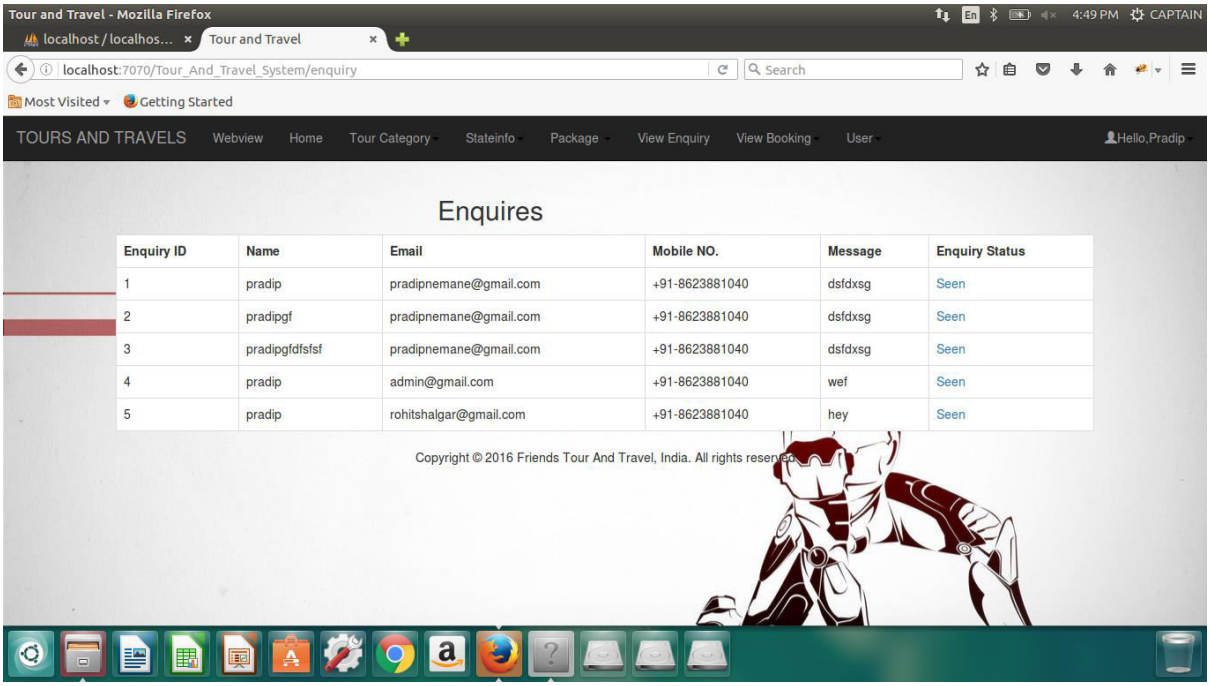
Add Category:



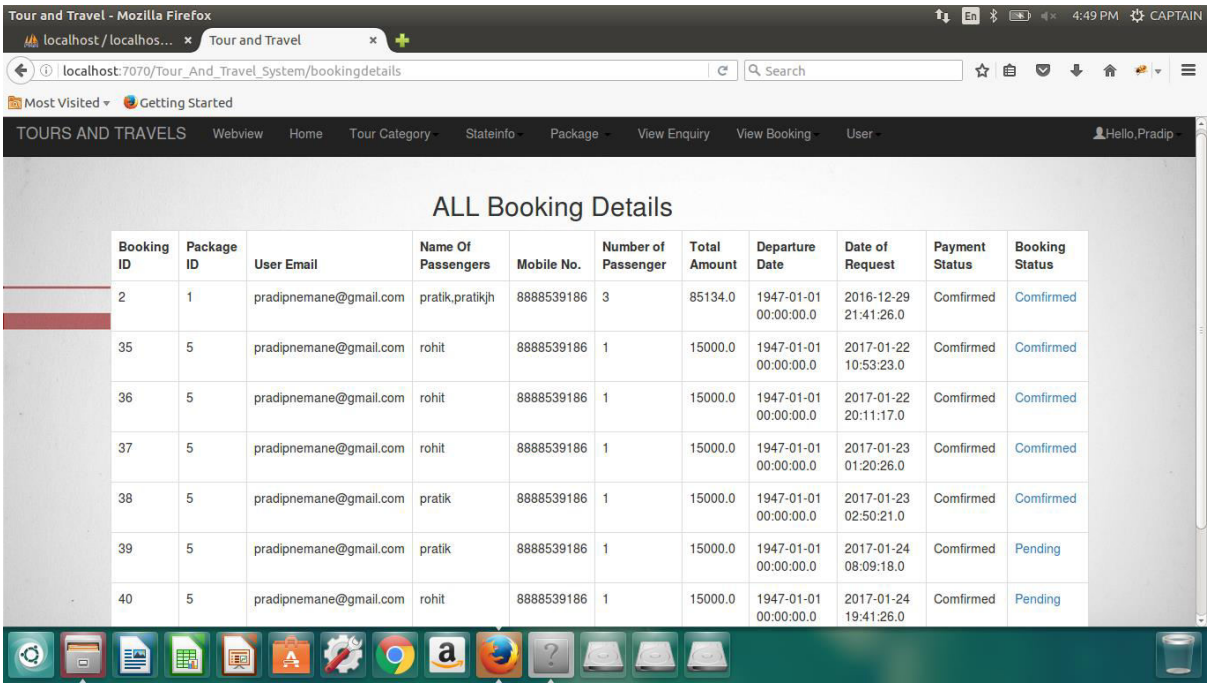
Add State:



View Enquiry:



View Booking:



BIBLIOGRAPHY

Core JavaTM 2 Volume I – Fundamentals 7 th Edition Pearson Education – Sun Microsystems

-Cay S. Hortsman

-Gary Cornell

Core JavaTM 2 Volume II – Advanced Pearson Education – Sun Microsystems

-Cay S. Hortsman

-Gary Cornell

Head First Servlets & JSP O'Reilly – SPD

-Eric Freeman

-Elisabeth Freeman

The Book of JavaScript 2 nd Edition SPD

-thau

Effective Java – Programming Language Guide Pearson Education – Sun Microsystems

-Joshua Bloch