

**EAST WEST UNIVERSITY**

# **Travelous**

**(An Information Portal for Tourists)**

*By*

Samia Bari

ID: 2011-3-60-027

*Supervised by*

Lecturer

Shakila Mahajabin Tonni

Department of Computer Science and Engineering

East West University

The Project has been submitted to the Department of the Computer Science and Engineering at East West University in the partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science and Engineering.

August, 2017

# Declaration

The Project has been submitted to the Department of the Computer Science and Engineering at East West University in the partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science and Engineering performed by me under supervision of Lecturer Shakila Mahajabin Tonni, Department of the Computer Science and Engineering at East West University. This is also needed to certify that the project work is under the course '**Project Work (CSE-497)**'. I, hereby, declare that this project has not been submitted elsewhere for the requirement of any degree or diploma or any other purposes.

Signature of the candidate

-----

**(Samia Bari)**

Dhaka, Bangladesh

Date: \_\_\_\_\_

# Abstract

The Travelous (Information Portal for Tourists) will facilitate local tourism industries and develop Bangladesh as an exciting tourist destination. It will assist tourists to find out new places and the available services for the tourists in the surrounding areas. Vendors will include information on their services in the portal. A user can be a contributor to the system also. (S) He can add a new place or add review to a place. As an additional feature, a user can indicate himself as a host and may provide rental services to the explorers who would prefer a home-like environment to stay than any commercial mean. This system allows user to easily get all the tourist information of a place. This application helps tourists to choose available tour offers that are going in a place. Here he can maintain an account. Vendors can sign up with an account and uploads their offer's detail and vendor detail to advertise their business. Tourists can be a host here and can serve the other tourists. Tourist can give his valuable reviews for a place which might help other tourists.

# Letter of Acceptance

This project is submitted by Samia Bari, Id: 2011-3-60-027 to the department of Computer Science and Engineering, East West University, Dhaka, Bangladesh is accepted as satisfactory for the partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science and Engineering on August 10, 2017.

## Board of Examiners

1. \_\_\_\_\_

Shakila Mahajabin Tonni

Lecturer

(Project Supervisor)

Department of Computer Science and Engineering

East West University, Dhaka, Bangladesh

2. \_\_\_\_\_

Dr. Md. Mozammel Huq Azad Khan

Professor

(Chairperson )

Department of Computer Science and Engineering

East West University, Dhaka, Bangladesh

# Acknowledgements

It has been a great pleasure for me to develop a Web Application on Information portal for Tourists called **Travelous**. I have gathered sufficient knowledge and experience during this project.

Most Importantly I would like to thank my honorable teacher and supervisor, Lecturer Shakila Mahajabin Tonni, Dept. of Computer Science and Engineering at East West University. Who guided me to proper analysis of the system and helped to develop an elegant and efficient system. It was a great pleasure to study and work with many gifted people who influenced us in many ways. First of all, I would like to thank my project supervisor Shakila Mahajabin Tonni. She encouraged me to seek out the clearest and deepest description of theoretical ideas as well as experimental findings. I am very grateful to her for her continuous support, advice and guidance.

Finally, I would like to convey my special thanks to my parents who have always given me tremendous support. Without their love, support and encouragement I would not had achieve this far.

# Table of Content

<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	MOTIVATION	1
1.2	OBJECTIVE	1
1.3	PURPOSE OF THE PROJECT	2
1.5	RELATED WORKS	3
1.6	USED TOOLS & TECHNOLOGIES	3
1.7	LIMITATIONS	3
1.8	STRUCTURE	4
<b>2</b>	<b>LITERATURE REVIEW</b>	<b>5</b>
2.1	KAYAK	5
2.2	TRIPADVISOR	6
2.3	MAKEMYTRIP	6
2.4	TRIVAGO	7
2.5	EXPEDIA.COM	8
<b>3</b>	<b>OVERVIEW OF PROPOSED SYSTEM</b>	<b>11</b>
3.1	REQUIREMENT ANALYSIS:	11
3.2	USE CASE DIAGRAM:	12
3.3	ACTIVITY DIAGRAM	16
3.4	CLASS DIAGRAM	23
3.5	DATABASE DESIGN	24
3.6	SOFTWARE REQUIREMENTS	29
3.7	SOFTWARE PLATFORM	30
<b>4</b>	<b>EVALUATION OF THE SYSTEM</b>	<b>33</b>
4.1	USER INTERFACE	33
<b>5</b>	<b>CONCLUSION AND FUTURE WORK</b>	<b>40</b>
5.1	CONCLUSION	40
5.2	FUTURE WORK	41

# List of Images

Fig 3.1: Use Case diagram for common user.	12
Fig 3.2: Use Case diagram for Admin	13
Fig 3.3: Use Case Diagram for Vendor	14
Fig 3.4: Use Case diagram for Tourists	15
Fig 3.5: Activity Diagram for Basic Search	16
Fig 3.6: Activity Diagram for Searching a specific place/vendor by name	17
Fig 3.7: Activity Diagram for Add a new Place	18
Fig 3.8: Activity Diagram for Editing a Place detail	19
Fig 3.9: Activity Diagram for User Sign Up	20
Fig 3.10: Activity Diagram for Adding a New Offer	21
Fig 3.11: Activity Diagram for Adding a new Hosting Offer	22
Fig 3.12: Class Diagram	23
Fig 3.13: Database Table of Place Information.	24
Fig 3.14: Database Table of Tourist Information.	25
Fig 3.15: Database Table for Host Information	26
Fig 3.16: Database Table for Vendor Details.	26

Fig 3.17: Table for Country	27
Fig 3.18: Table for City.	27
Fig 3.19: Table for Offer Information.	27
Fig 3.20: Table for Offer Type.	28
Fig 3.21: Table for Host Type.	28
Fig 3.22: Table for Place Pictures	28
Fig 3.23: Table for Vendor Picture	29
Fig 3.24: Table for Vendor Type	29
Fig 4.1: Home Page	33
Fig 4.2: Search Form	34
Fig 4.3: Results after searching a Travel spot.	34
Fig 4.4: As admin Log In page	35
Fig 4.5: Admin Panel page after logged in	35
Fig 4.6: Vendor Sign Up Page.	36
Fig 4.7: Tourist Sign Up page.	37
Fig 4.8: Form for adding a new Place	38
Fig 4.9: Form for adding a new hosting offer	39
Fig 4.10: Host List after searching a place	39



# 1 Introduction

A portal is a website that acts as a multi sources or single source for all information on a specific domain. So this Information portal for Tourists will be a web portal which offers the tourists a broad array of information of tourist spots, attractions, cultures, and about the related vendors, arranged in a way that is most convenient for the tourist to access and receive the service of knowledge and adventure. It is designed, implemented in a way that it can become the starting or entry point of a tourist introducing him into various information, resources and other sites in the internet about tour. It has the facility for tourists to check and search the different tourist spots around the world, different vendors including hotels, travel agents, tour guide and restaurants. Here a tourist can hold an account give his reviews about a place or add a place as a tourist attraction, track his adventure diary and be a host as well. Here vendors also can have an account and give their special offers to the tourists and let tourists know about their facilities. The main aim of this web site is to create a gateway for tourists where they can come and design their trip.

## 1.1 Motivation

In our country there are many tourist spots but not all of them are very popular or known to all even though they are very attractive and historically nourished or naturally beautiful. There are very few websites that gives a tourist a whole bunch of knowledge and travel facility information all together. That's why I considered the information portal for tourists as my project. This website will give a tourist so many information about our country's tourist spots, help him to manage his tour and make the tour more enjoyable and easier.

## 1.2 Objective

### ❖ Primary Objectives:

- To increase the efficiency and services to the tourists and vendors through better application of technology in their respective tasks.
- To be able to stand from competitors in the portal sites.

❖ **Specific objectives:**

- To enable the search option for the tourists to search any place he wants to visit and get the information about it.
- To enable an option for the vendors to upload their offers and let the tourists know about them.
- To enable the tourists to have a visualization of the spots before planning the tour.
- To enable the tourists to maintain a travel diary.

### **1.3 Purpose of the project**

Before going to a place, a tourist has to know many things about that place and the scenario of that place's life style. So that he could plan a tour with all necessary information and enjoy the adventure without any complexity. This website will give him almost all the necessary information that he might need before going to a tour. Here a tourist can search a place or search a country for its interesting tourist spots, make a travel diary, give review about a place. A vendor can add offers, add places as a tourist spot. An admin can keep track of all accounts and handle security options through admin panel.

### **1.4 Project scope**

The information portal tourists will be a web based application which will be created by using .Net Platform. Its main aim is to simplify and improve the efficiency of the tour manager for both tourists and vendors. Minimize the time consumption for search or any information related to tour. A tourist can see categorized spots with their full information of a city or country. A tourist can also be a host and have a visual confirmation of his hosting.

## 1.5 Related Works

There are some travel websites that gives almost similar facilities to the tourists. Some gives information of tourist spots and some of them gives the hotel or restaurant reservation, offer accepting ability and travel agent's information. The Kayak is one of them. It allows tourists to find cheap flight and package deals from an array of travel websites, it also notifies when any decrease in tourist's preferred journey fare. The SkyScanner is a travel search engine, which gives facility to check for flights, hotels and car rentals from providers across the board. The Expedia helps traveler to book their flights, tour packages, hotels, car rentals, cruises, in-destination activities and other tourist services. <sup>[1]</sup> The TripAdvisor provides reviews of travel-related content including interactive travel forums. There are some other websites with same objectives like, MakeMyTrip, ClearTrip, Musafir etc.

## 1.6 Used Tools & Technologies

To build up this Information Portal, the main language of programming C# on .Net platform using Entity Framework 6.0 in MVC 5 methodology is used. It is a database first project where SQL server is used to maintain system database. For designing the view for both front end and backend, I have used HTML, CSS.

- Programming Language: C#
- Framework: .Net, Entity Framework 6.0
- Database system: MS SQL server 2012, MS SQL Server 2017
- Markup languages: HTML, CSS
- IDE: Visual Studio 2015

## 1.7 Limitations

- Requires internet connection and the person must be computer literate.
- It may not show the complete details of the users after accepting an offer or service. Sometimes system falls.
- For booking purpose or receiving any service tourist have to directly go to vendor's website.

## 1.8 Structure

The book is structured as follows:

In Chapter 1, There is a short overview of the project topic, its scope, my purpose of creating the project on this topic and the technology that are used to create this portal.

In Chapter 2, Contains the review of other websites that are created on project topic (Information Portal for Tourists). Their facilities and how they have been created.

In Chapter 3, Has the detail description about the project topic. how it is designed, its work flow, user category detail, database format, detail of the technologies that have been used.

In Chapter 4, It is the place where the created website has been evaluated by showing different types of data. Here the functionalities of Travelous has been shown.

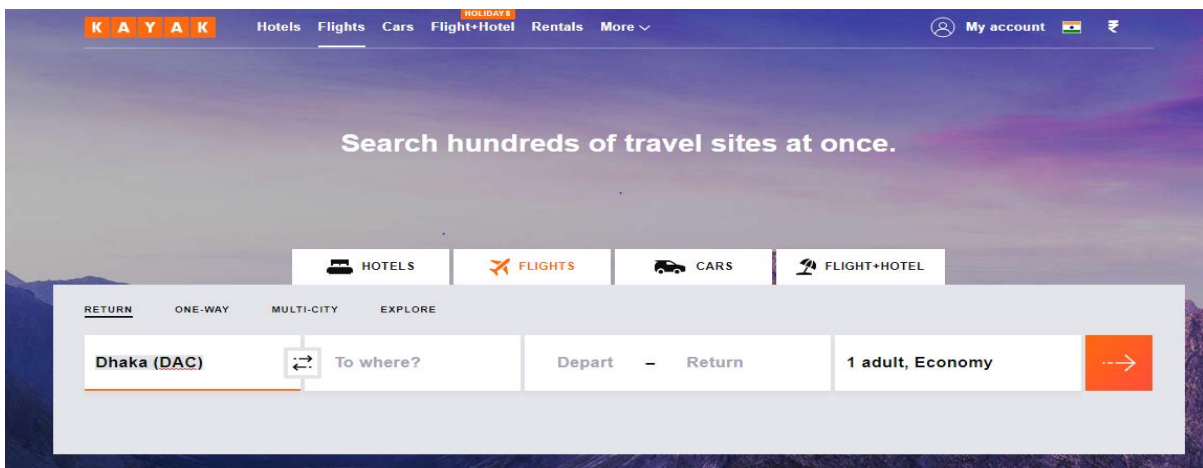
In Chapter 5, Has been designed with the conclusion and future works. Here conclusion includes what I have learned by working on this project, what can be done in future to make the Travelous a nourished and an excellent website.

## 2 Literature Review

Before going on a vacation a tourist or a person has to check many things about the place he is going to visit and he needs many facilities and services like accommodation, food, transport, hospital facility, banking facilities and above all the fare system of any service he is going to take at that place, so that he can travel without any tension. So there are some travel websites that gives the information of those facilities and services provider's information and detail of their offers. These travel websites give the information of tourist spots and some of them gives the hotel or restaurant reservation, offer accepting ability and travel agent's information. Some of these are only specialized in hotel information giving and some are only in Tourist spots or Travel agents information giving.

### 2.1 KAYAK

Kayak searches hundreds of travel sites at once from online travel agencies to airlines, hotel and rental car websites and gives the information a tourist need to make the right decisions for a tour. It is one of the leading travel websites.<sup>[3]</sup> It is a fare aggregator and travel metasearch engine operated by The Priceline Group.

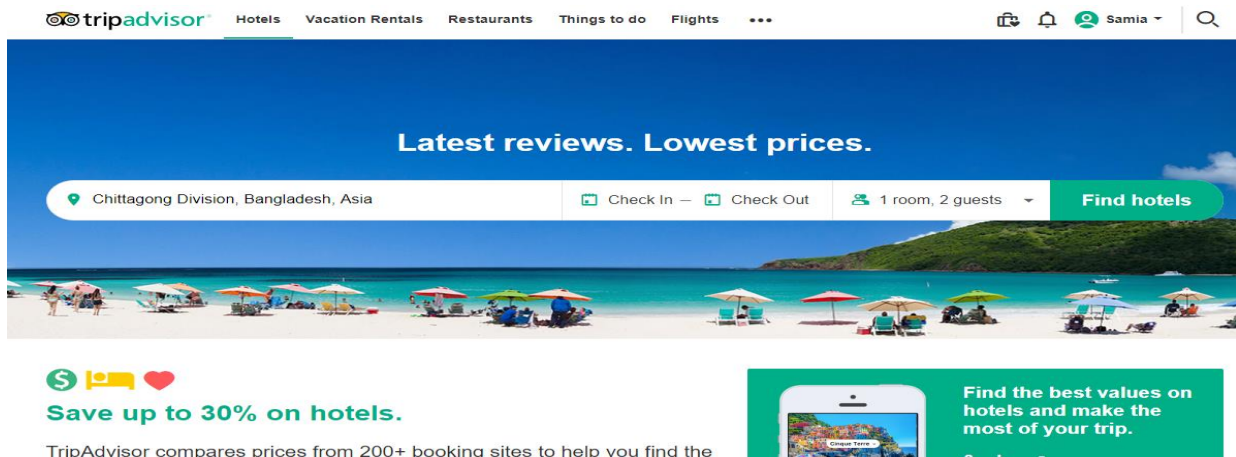


Unlike other websites it doesn't sell anything but works as a tool for travelers which searches other websites and shows their offer in a brief. It also has a mobile app which does the same as the website. KAYAK is available in over 40 international sites and 21 languages. Each

year they process 1.5 billion queries for travel information. <sup>[4]</sup>To receive the facility of this website people has to visit <https://www.kayak.co.in>.

## 2.2 TripAdvisor

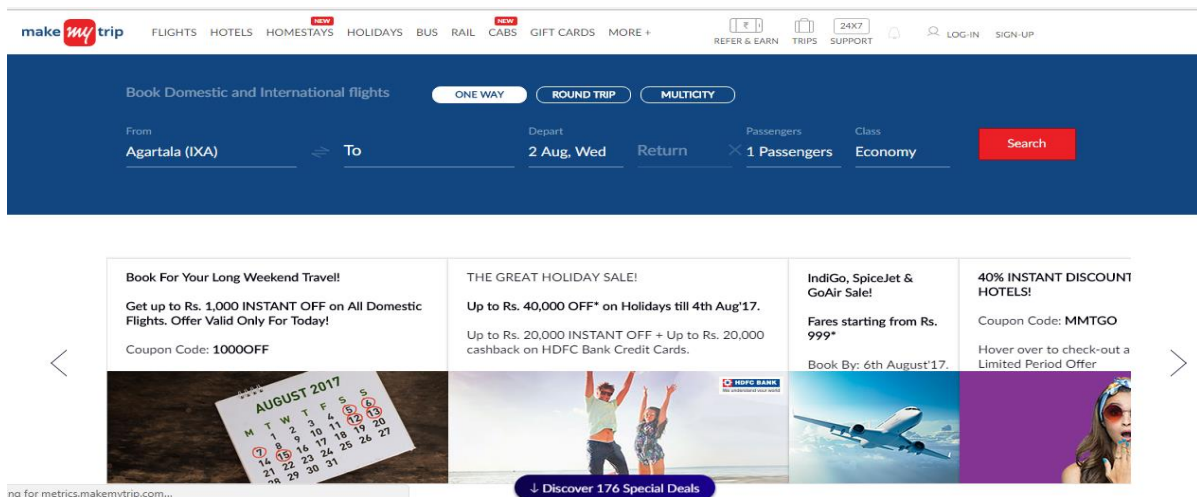
TripAdvisor, the world's most visited travel website. It is an American travel website company providing hotels booking as well as reviews of travel-related content. It also includes interactive travel forums. <sup>[4]</sup> TripAdvisor is an early adopter of user-generated content. Its address is <https://www.tripadvisor.com>. The website services are free to users, who provide most of the content, and the website is supported by a hotel booking



facility and an advertising business model. In its site there are more than 500 million reviews and opinions covering the world's largest selection of travel listings worldwide over 7 million accommodations, airlines, attractions, and restaurants. TripAdvisor also compares prices from more than 200 hotel booking sites so travelers can find the lowest price on the hotel that's right for them.

## 2.3 MakeMyTrip

MakeMytrip is an Indian online travel company founded in 2000. provides online travel services including flight tickets, domestic and international holiday packages, hotel reservations, rail and bus tickets based on India. The company has been recognized as one of India's best travel portals. <sup>[5]</sup> After Completing the domestic target it has also reached to the US market. Its address is <https://www.makemytrip.com>.



The company also operates through 65 retail stores across 50 cities in India, along with offices in New York City and Sydney.

This website has been created to empower the Indian travelers with instant booking and comprehensive choices of hotels and travel agents.

## 2.4 Trivago

The famous website <https://www.trivago.com> is a website which is specialized in internet-related services and products in the hotel, lodging and metasearch fields. Trivago N.V., often referred to as Trivago, is a German multinational technology company. [6] It was the first hotel search engine in Germany, and is one of the fastest growing companies in Germany.

Find your ideal hotel  
for the best price

It only gives a tourist the information about any hotels of any place, their pricing detail. A tourist can search any hotel according to his budget at this field. Trivago also has some interesting features like Trivago Hotel Manager, which allows hoteliers to manage their presence on the Trivago site. The Hotel Manager Pro, which enables hoteliers to update and customize their own listings, as well as access analytics about competitor rates.

## 2.5 Expedia

Expedia.com is a travel website owned by Expedia Inc. The website can be used to book airline tickets, hotel reservations, car rentals, cruises, vacation packages and various

The screenshot displays the Expedia.co.in website interface. At the top, there is a yellow banner with a 'NEW' tag and a promotion: 'Expedia+ Get an extra 10% off or more on select hotels with Member Pricing. Join now, it's free!'. Below the banner, the website header includes the Expedia.co.in logo, navigation links (Home, Hotels, Flights, Discover, Car Rental, Holiday Activities, Last Minute Deals, Deals), and user account options (Account, My Scratchpad, Manage Trips, Support, Sign In, Create an Account). The main search area features a 'Going to' field with a placeholder 'Destination, hotel name, airport, train station, landmark or address', and a search form with fields for 'Check in' (dd/mm/yyyy), 'Check out' (dd/mm/yyyy), 'Rooms' (1), 'Adults (18+)' (2), and 'Children (0-17)' (0). A yellow 'Search' button is positioned below the search form. At the bottom of the search area, there is another yellow banner: 'Subscribe for Member Pricing - Unlock an extra 10% or more off select hotels.' The background of the search area shows a tropical beach scene with palm trees and a sunset.

attractions and services via the internet or telephone agents. The site uses multiple global distribution systems like Amadeus or the Sabre reservation systems for flights and for



hotels, World span and Pegasus, along with its own hotel reservation system for contracted, bulk-rate reservations. Here a tourist can search for a place and can explore what he can do there by using the thing to do feature. To check out the facility a person has to visit <https://www.expedia.co.in>.

## 2.6 Reviews

Reviews.com only deals with reviews taken from the travelers about any hotel or travel agency or about their given offers to the tourists. To get the review a person has to go to <http://www.reviews.com/travel-sites>. Tourists can come here search for reviews for any place which will help them to decide their visit to that place. It also contains the reviews about travel websites like TripAdvisor, Expedia etc.



# Travel Site Reviews

🕒 Latest Update June 30, 2017

## The Best Travel Sites

A vacation is something you want to make sure you're paying for exactly what you get, and that means booking through a website you trust. There are 15 standout companies, especially [Booking.com](#) with perks like secret deals, a smooth booking process, and discounts on places like vacation rentals and hostels.

### 🏆 Best Overall Site

[Booking.com](#)

By providing robust tools that greatly

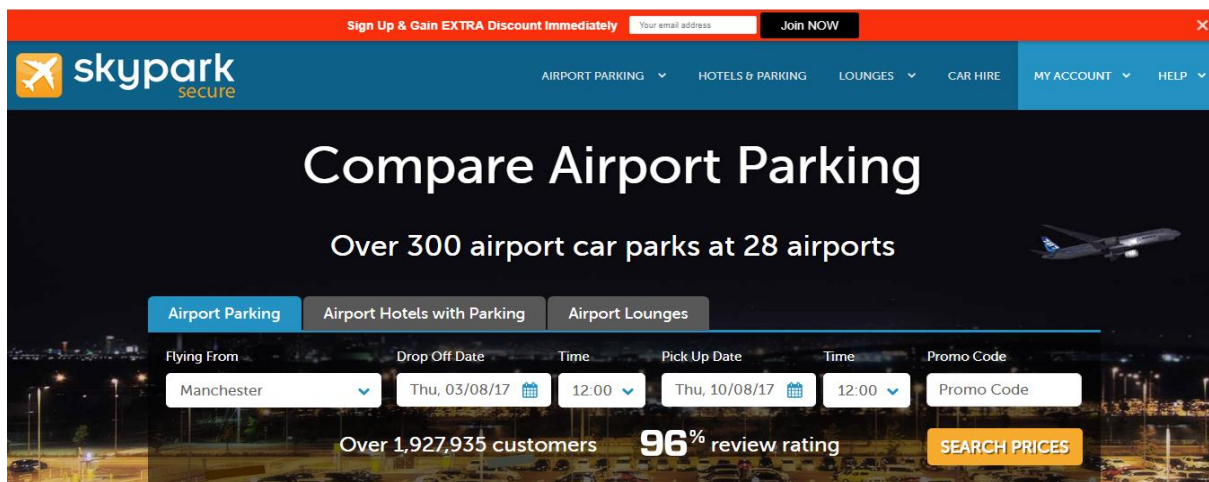
Want to be the first to know everything?  
[Join our newsletter.](#)

Subscribe

Plus, instant access to our exclusive guide: "Make the Right Choice: A 10-Minute Guide to Not Messing Up Your Next Purchase".

## 2.7 SkyParkSecure

SkyParkSecure is an online travel company offering airport parking at all UK airports. Earlier the web site was a simple booking system, enabling customers to search by a range of dates



and display the results for the one product on offer. The site was quickly developed over the coming years to integrate the rapid addition of parking products across the UK. <sup>[11]</sup> This also enabled customers to choose from a wide range of airport parking services, but also enabled staff within the company to manage many of the customers' needs within the system. To take the facility a person has to visit <https://www.skyparksecure.com/> to their website.

# 3 Overview of Proposed System

Development of web based system requires analysis of the process to be digitized in order to enable a correct system, a system that functions as required and to assist the potential users of the system understand the general functionality of the system. The analysis specifies the system's objectives and constraints to which designers have to comply. The purpose of doing analysis is to transform the system's major inputs into structured specification. This website has four requirement sections. The admin section, common user section, registered tourist section and the vendor section.

## 3.1 Requirement analysis

- ❖ Each user type vendors or tourists need to register himself/herself a registered user for accessing necessary information. They also have a user password and email. They can log in to the system by using their email and password.
- ❖ Admin need to log in to the system to operate the system. Admin has an individual name or unique log in user email and password. Through this email and password admin can log in to the system.
- ❖ Admin can update all places information, block a place from the access of the other users. He/she can insert a new place to visit.
- ❖ Admin can see the information of the vendors including account holder's information and block them or their any offer if necessary.
- ❖ Admin can see all information of a tourists including their emergency information and can block them if needed.
- ❖ A common user can only search for a place, or vendor and see their address, likes and dislikes.
- ❖ A tourist can add a new place, insert pictures for that place.
- ❖ A tourist can become a local /international host and add his offers. He/she can update her hosting and account information as well.
- ❖ A vendor can add offers for his company, add a new place, update account and offer information.
- ❖ Admin can see any blocked accounts and places or expired offers.

## 3.2 Use Case Diagram

### 3.2.1 Use Cases for Common user

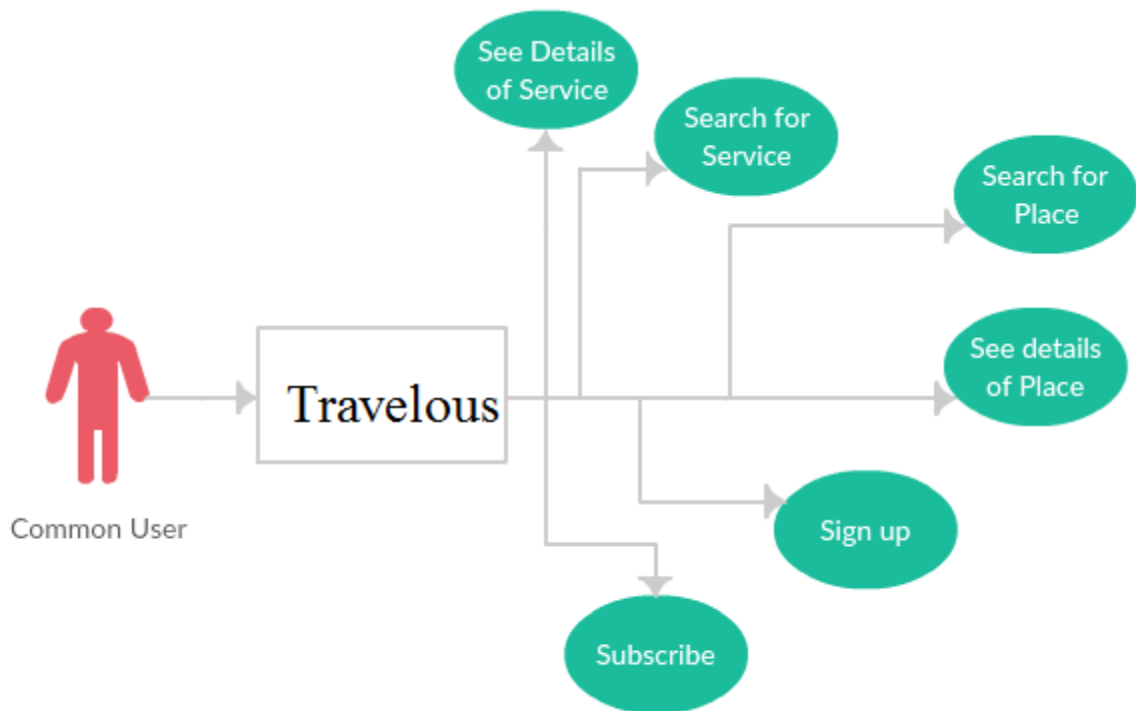


Fig 3.1: Use Case diagram for common user.

#### Functions provided:

- **Search place:** A common non registered user can only search for a place with country and city name.
- **Search Services:** Non registered user only can search for a service by choosing country name and city name and giving his/her visit time.
- **Subscribe:** Non-registered user can subscribe to newsletter to for getting email about new places.
- **See Service/ Place detail:** common user can check the partial detail of service and place by searching.
- **Sign Up:** A common user can become a regular user by signing up.

### 3.2.2 Use Cases for Admin user

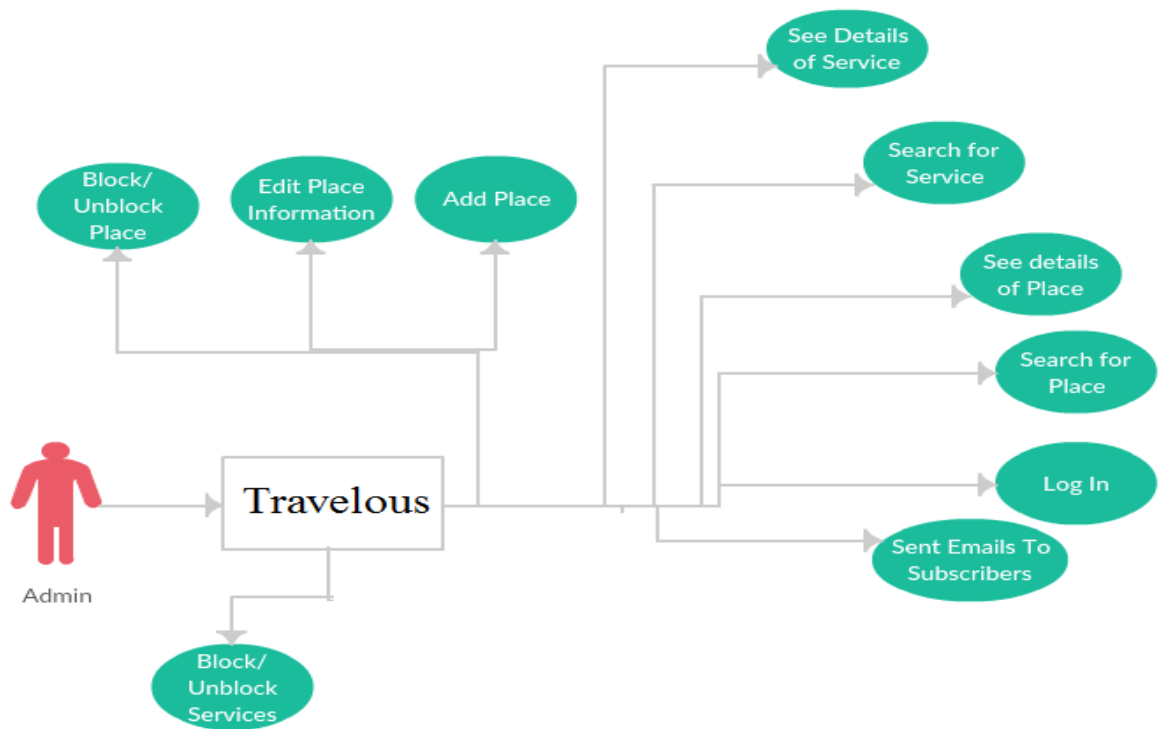


Fig 3.2: Use Case diagram for Admin

#### Functions provided:

- **Search a Place:** Admin can search for a place with country and city name
- **Search Services:** Admin can search for a service by choosing country name and city name and by giving any time period
- **See Service/ Place detail:** Admin can check the full detail of service and place by searching.
- **Block/Unblock User/Service:** Admin can block/unblock any user/service.
- **Send Email:** Admin can email to the subscribers for new places.
- **See Tourist/ Vendors Detail:** Admin can see full detail of any users.
- **Add Place:** Admin can add any place for tourist attraction.
- **Edit/Block Place:** Admin can block/ edit place information if needed.

### 3.2.3 Use Cases for Vendors

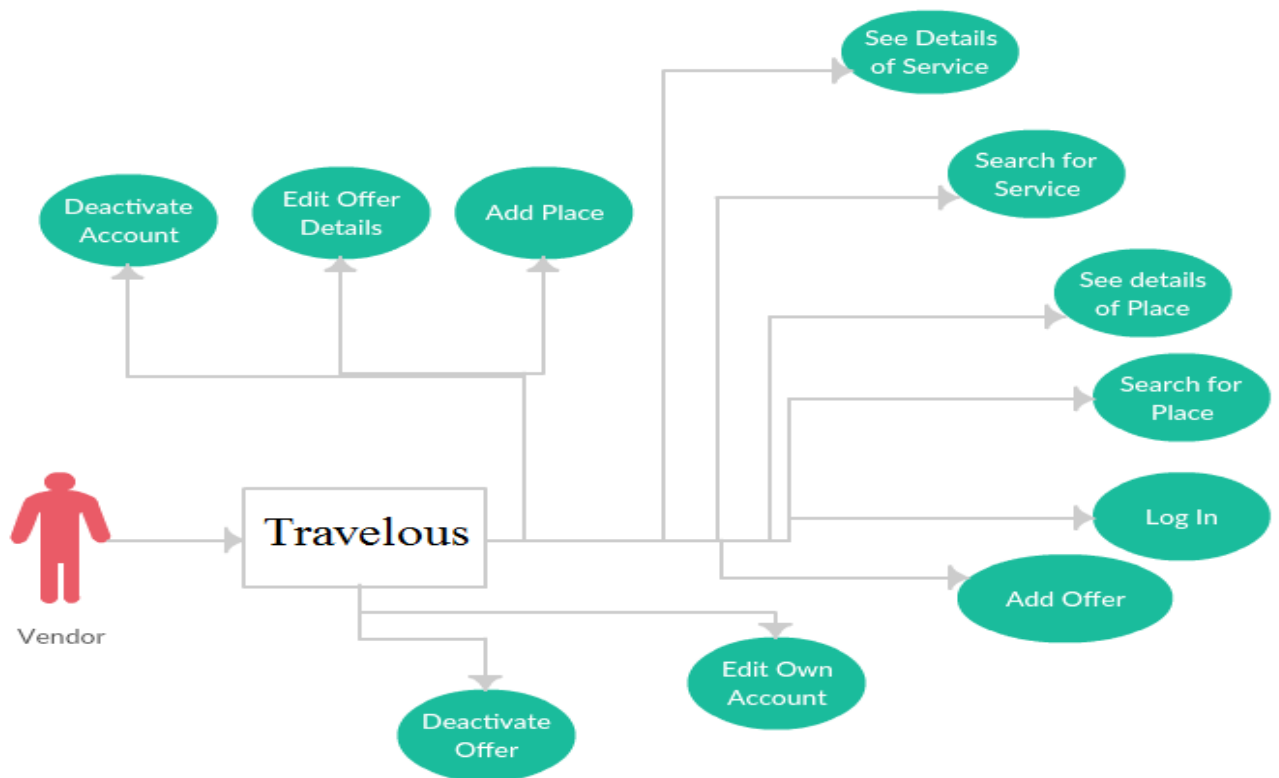


Fig 3.3: Use Case Diagram for Vendor

#### Functions provided:

- **Search place:** Vendor can search for a place with country and city name.
- **Search Services:** A Vendor can search for other services by choosing country name and city name and by giving any time period.
- **See Service/ Place detail:** Vendor can check the partial detail of other services and full detail of a place by searching.
- **Add Offer:** Vendor can add an offer for tourists.
- **Edit Offer:** Vendor can edit his own offers.
- **Edit Account:** Vendor can edit his/ her business and account information.
- **Add A Place:** Vendor can a new place.

### 3.2.4 Use Cases for Tourists

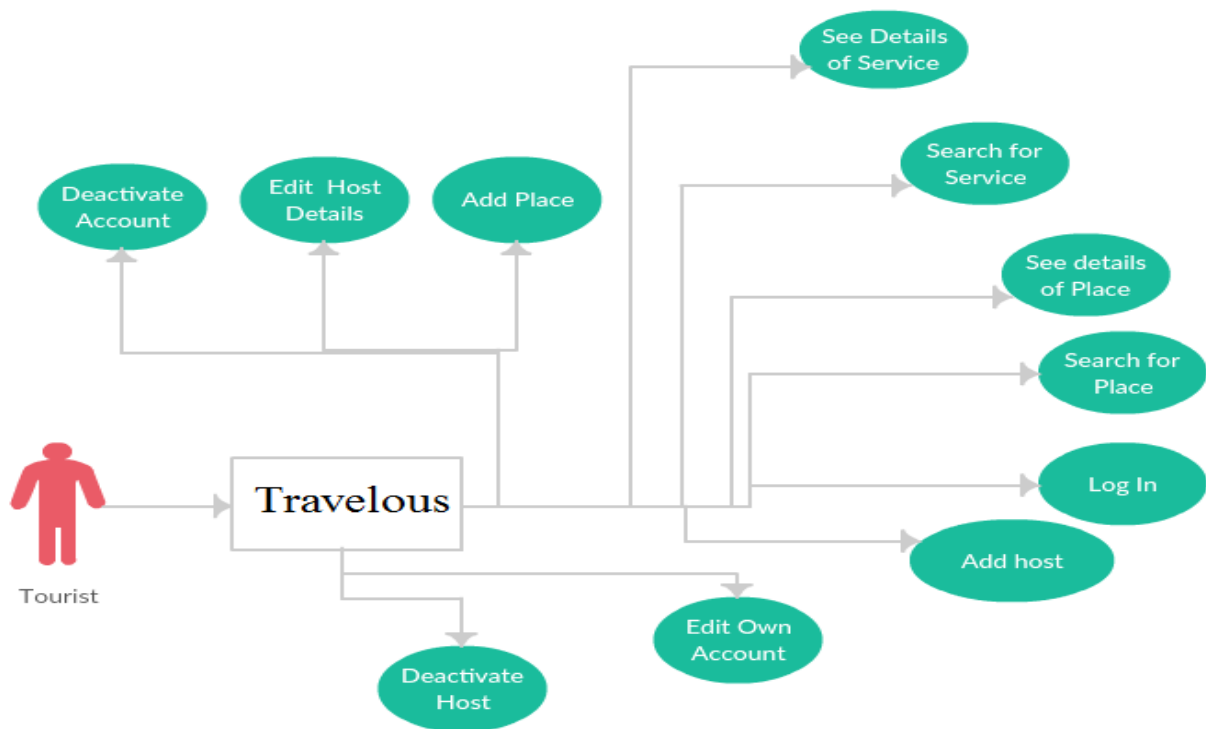


Fig 3.4: Use Case diagram for Tourists

#### Functions provided:

- **Search place:** Tourist can search for a place with country and city name.
- **Search Services:** A tourist can search for other services by choosing country name and city name and by giving any time period.
- **See Service/ Place detail:** Tourist can check the necessary detail of services and full detail of a place by searching.
- **Be a Host:** Tourist can be a host for other tourists.
- **Edit Hosting:** Tourist can edit his hosting details.
- **Edit Account:** Tourist can edit his/ her emergency and account information.
- **Like/Dislike a Place/Service:** Tourist can like or dislike any place or service.
- **Add a new place:** Tourist can add a new place as a tourist spot.

## 3.3 Activity Diagram

### 3.3.1 Activity diagram for Basic Search

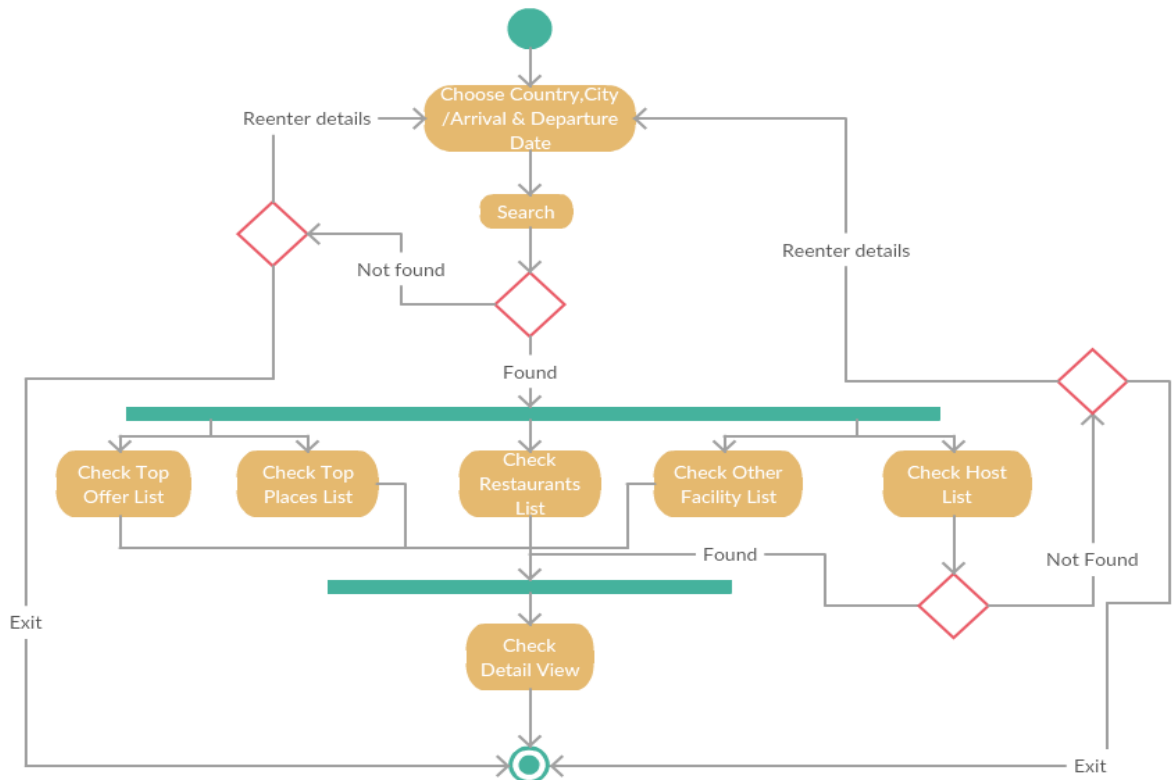


Fig 3.5: Activity Diagram for Basic Search.

#### Activity Flow Description:

This is a common Feature for registered or nonregistered user.

- A person has to Choose a Country/City where he/she wants to visit, here can he/she can also add his/her Arrival and Departure time for additional search.
- If there is any value found, then system will give him/her a list of searched vendors and hosts and place of that Country and City.
- From there the person can see the whole list of selected category and get the detail of selected Place/offer/Vendor/Service/Host.



### 3.3.2 Activity Diagram for Specific Search a Place/Vendor

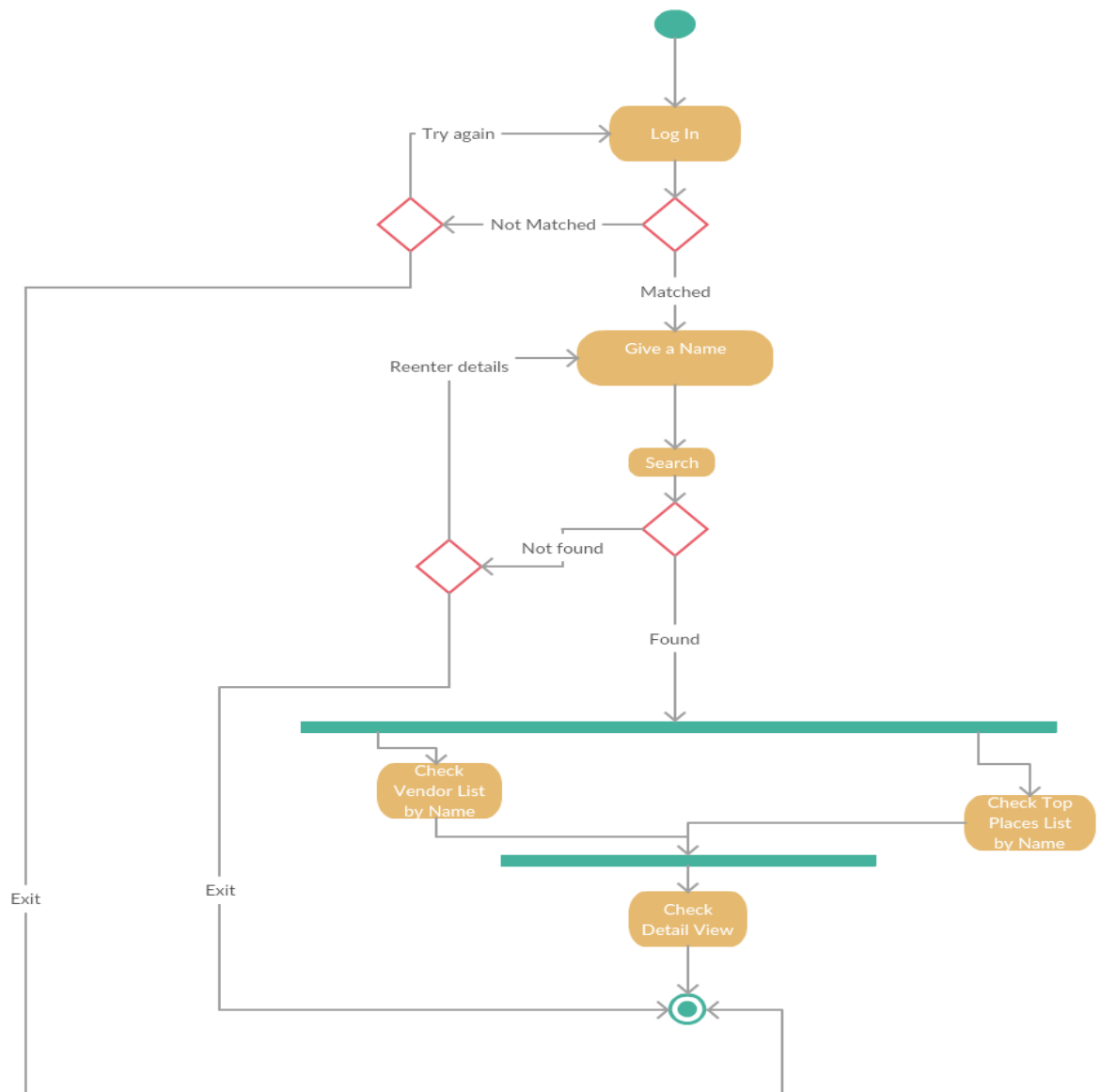


Fig 3.6: Activity Diagram for Searching a specific place/vendor by name.

#### Activity Flow Description:

This feature is only open for a registered Tourist or system Admin.

- A Tourist/Admin has to log in first, then he/she have to give a name for place for the place/vendor he/she want to search.
- If there is any value found, then system will give him/her a list of vendors and places with his given name.
- From there user can select a place or vendor for the detail view.

### 3.3.3 Activity Diagram for Adding a new Place

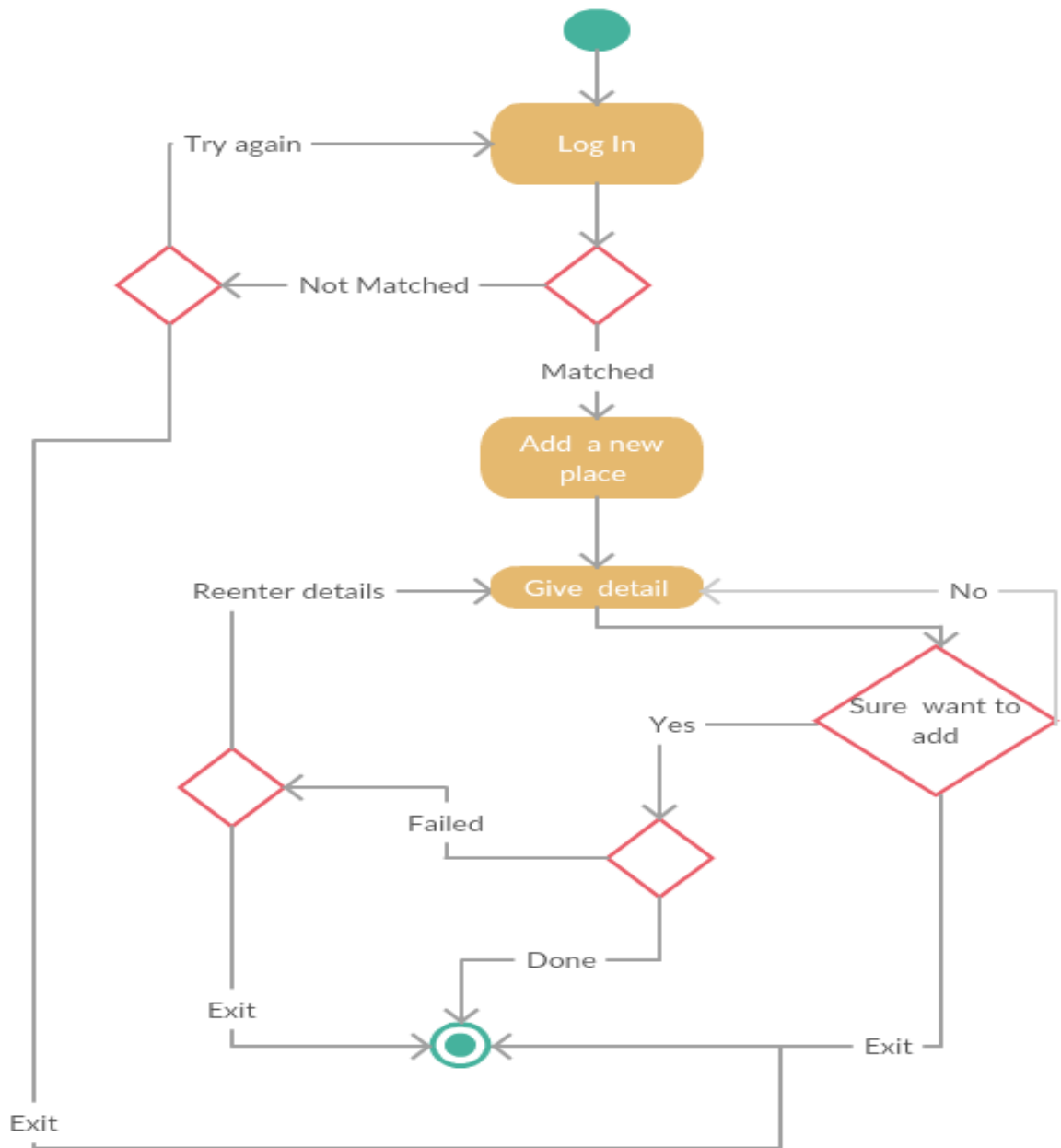


Fig 3.7: Activity Diagram for Add a new Place.

#### Activity Flow Description:

This feature is only open for registered users (Vendor/Tourist) and System Admin.

- User has to log in, then he/she has to choose add a new place option from the action list.
- He/she has to give the full detail of the place which she/he wants to add.
- Then has to verify if he/she wants to add the place or not.

### 3.3.4 Activity Diagram for Editing a place

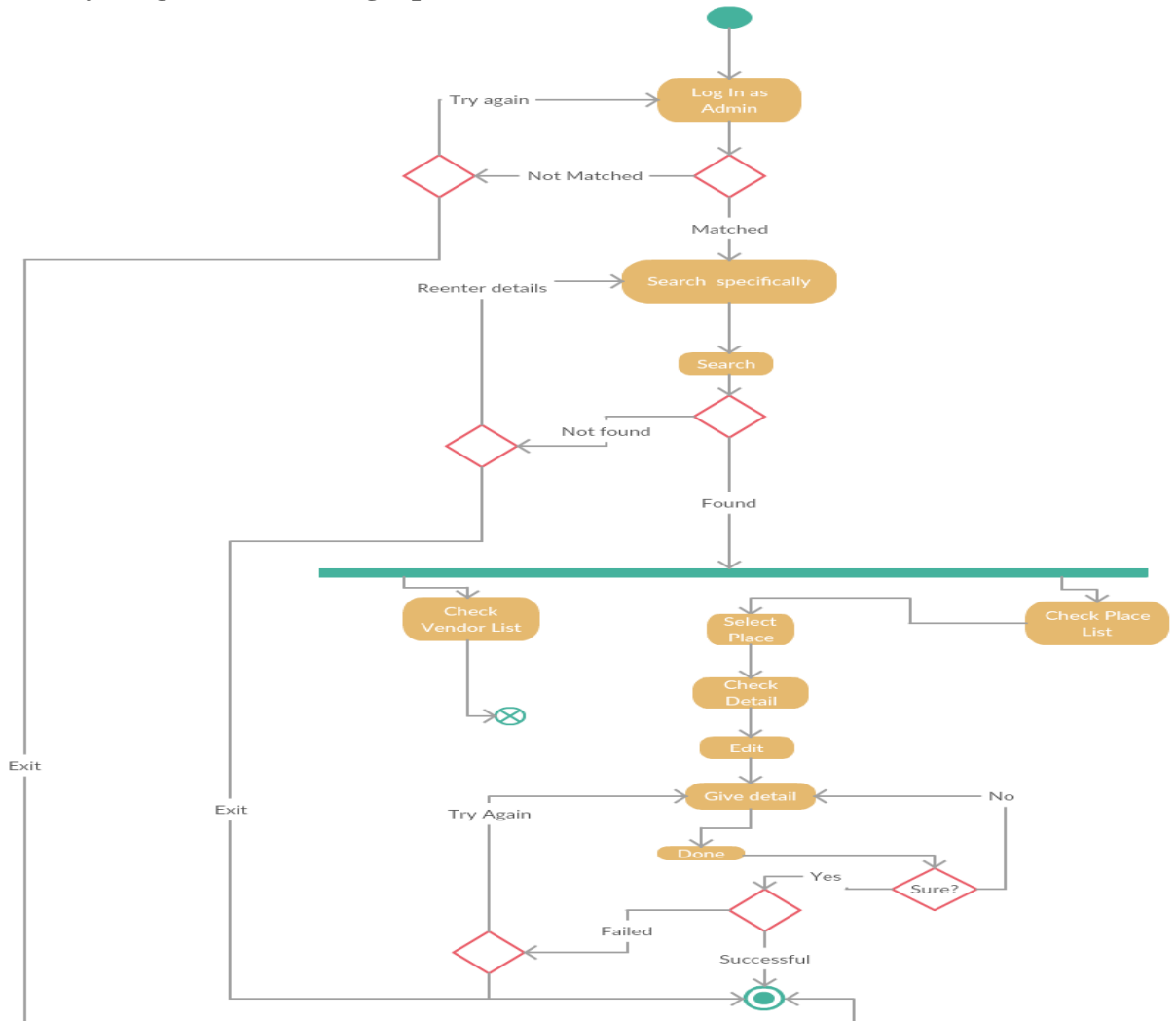


Fig 3.8: Activity Diagram for Editing a Place detail.

#### Activity Flow Description:

This feature is only open for the admin.

- Admin has to log in first to access the action.
- Then he/she has to search a specific place by its name.
- From the place list he/she has to select the place for detail view and select the edit option.
- Admin has to give the detail for edit and decide if he/she wants to edit or not.

### 3.3.5 Activity Diagram for User Sign Up

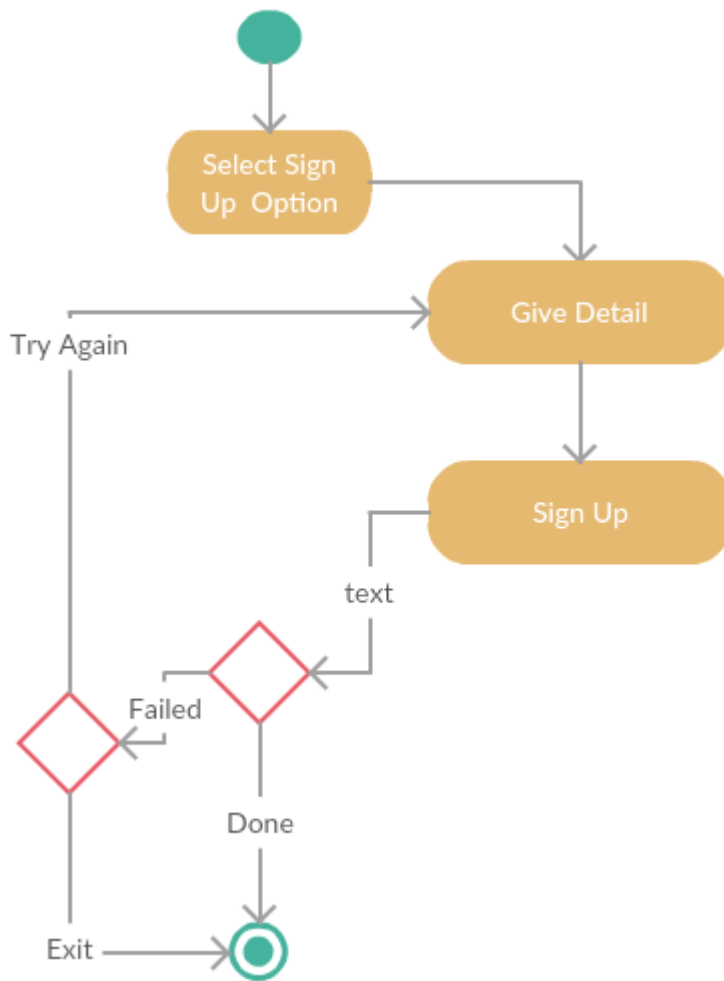


Fig 3.9: Activity Diagram for User Sign Up.

#### Activity Flow Description:

- A person has to select a sign up option whether he/she wants to be a vendor/regular tourist.
- Then she/he has to give a required detail for signing up.
- If given detail is valid then the sign up will be done otherwise person has to try again.

### 3.3.6 Activity Diagram for Add Offer

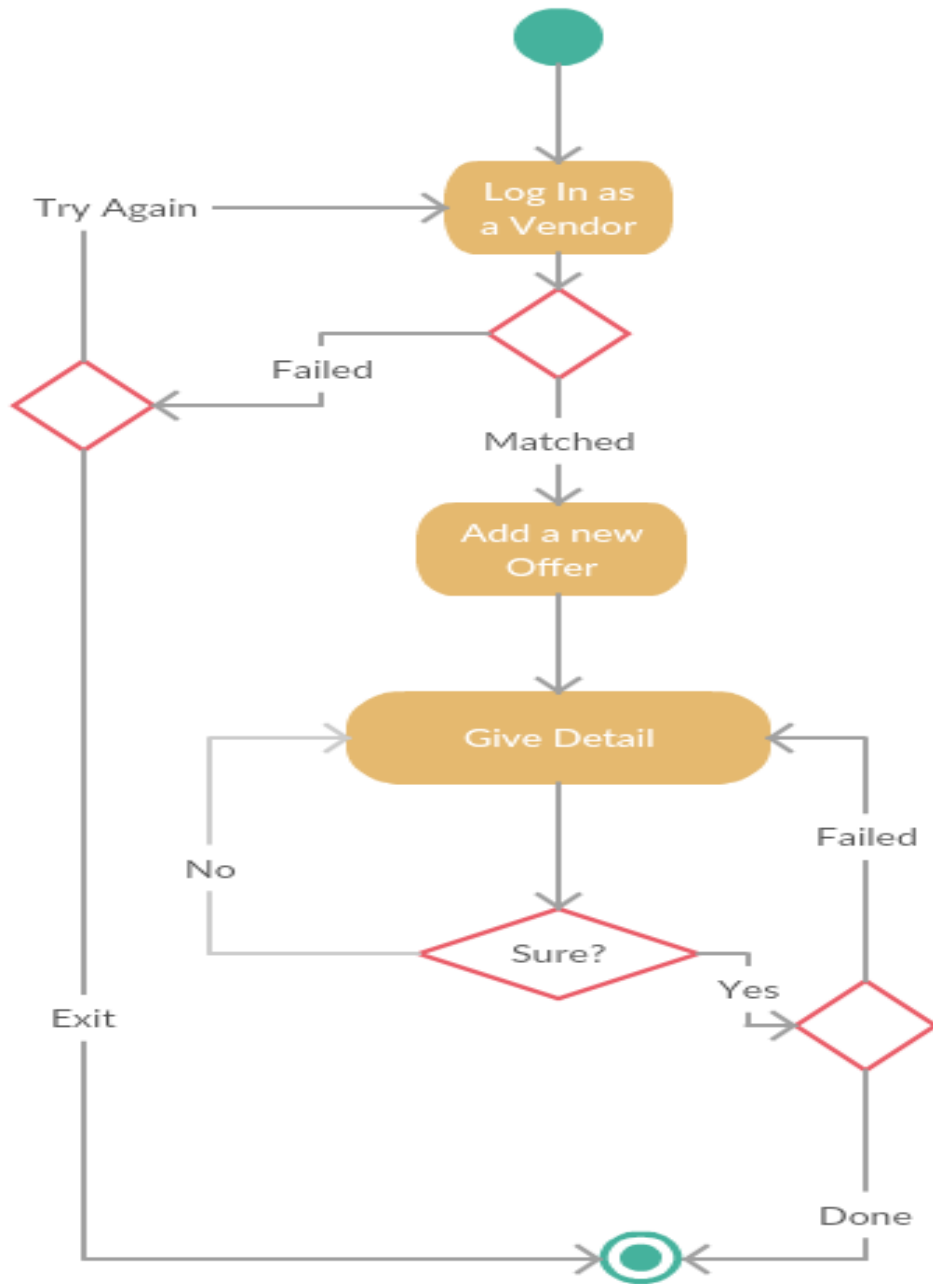


Fig 3.10: Activity Diagram for Adding a New Offer.

#### Activity Flow Description:

This option is only open for a Vendor.

- User has to select the Add a new offer option from the user home.
- Then he/she has to fill the whole detail of the offer.
- If there is no problem, then the addition will be successful otherwise user has to try again.

### 3.3.7 Activity Diagram for To be a Host

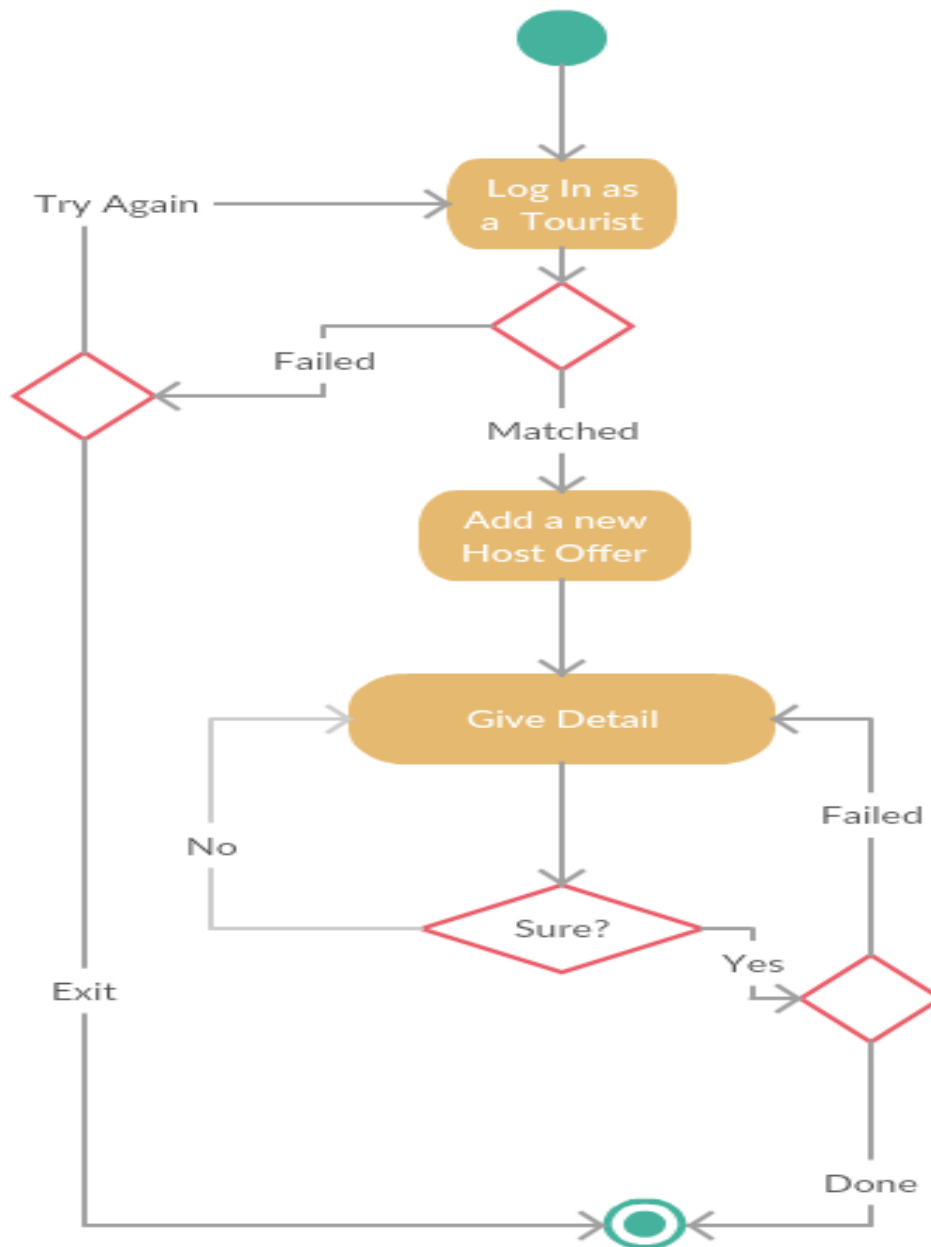


Fig 3.11: Activity Diagram for Adding a new Hosting Offer.

#### Activity Flow Description:

This option is only open for a Tourist.

- User has to select the Add a New Host Offer option from the user home.
- Then he/she has to fill the whole detail of the offer.
- If there is no problem, then the addition will be successful otherwise user has to try again.

### 3.4 Class Diagram

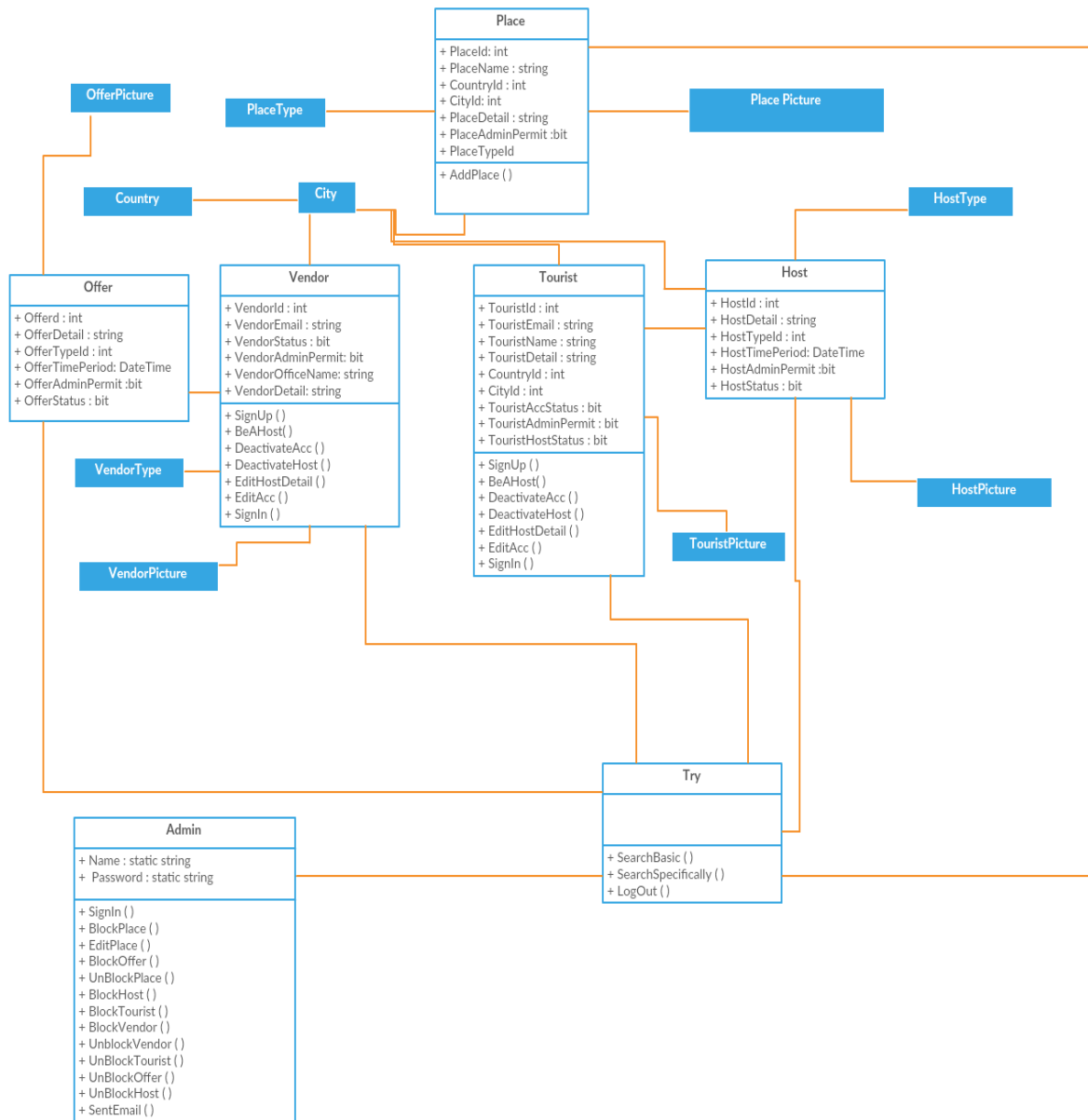
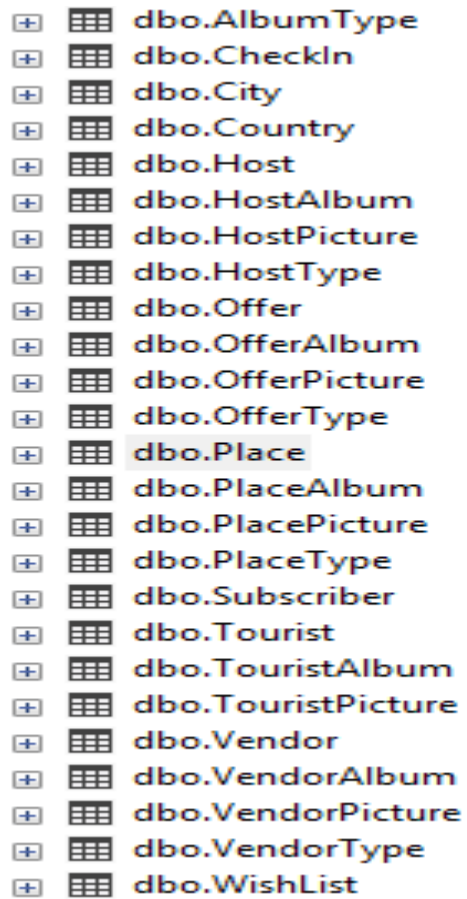


Fig 3.12: Class diagram for Travelous.

## 3.5 Database Design

### 3.5.1 Database Table List



The image shows a vertical list of database tables. Each entry consists of a small square icon with a plus sign, a small grid icon, and the table name. The table names are: dbo.AlbumType, dbo.CheckIn, dbo.City, dbo.Country, dbo.Host, dbo.HostAlbum, dbo.HostPicture, dbo.HostType, dbo.Offer, dbo.OfferAlbum, dbo.OfferPicture, dbo.OfferType, dbo.Place, dbo.PlaceAlbum, dbo.PlacePicture, dbo.PlaceType, dbo.Subscriber, dbo.Tourist, dbo.TouristAlbum, dbo.TouristPicture, dbo.Vendor, dbo.VendorAlbum, dbo.VendorPicture, dbo.VendorType, and dbo.WishList. The 'dbo.Place' entry is highlighted with a light gray background.

+ [grid]	dbo.AlbumType
+ [grid]	dbo.CheckIn
+ [grid]	dbo.City
+ [grid]	dbo.Country
+ [grid]	dbo.Host
+ [grid]	dbo.HostAlbum
+ [grid]	dbo.HostPicture
+ [grid]	dbo.HostType
+ [grid]	dbo.Offer
+ [grid]	dbo.OfferAlbum
+ [grid]	dbo.OfferPicture
+ [grid]	dbo.OfferType
+ [grid]	dbo.Place
+ [grid]	dbo.PlaceAlbum
+ [grid]	dbo.PlacePicture
+ [grid]	dbo.PlaceType
+ [grid]	dbo.Subscriber
+ [grid]	dbo.Tourist
+ [grid]	dbo.TouristAlbum
+ [grid]	dbo.TouristPicture
+ [grid]	dbo.Vendor
+ [grid]	dbo.VendorAlbum
+ [grid]	dbo.VendorPicture
+ [grid]	dbo.VendorType
+ [grid]	dbo.WishList

Fig 3.13: Database Table List.



### 3.5.2 Database Table for Place Information

Column Name	Data Type	Allow Nulls
PlaceId	int	<input type="checkbox"/>
PlaceName	varchar(MAX)	<input checked="" type="checkbox"/>
CountryId	int	<input checked="" type="checkbox"/>
CityId	int	<input checked="" type="checkbox"/>
PlaceTypeId	int	<input checked="" type="checkbox"/>
PlaceAddress	varchar(MAX)	<input checked="" type="checkbox"/>
PlaceDetail	nvarchar(MAX)	<input checked="" type="checkbox"/>
PlaceAdminsPermit	bit	<input checked="" type="checkbox"/>
PlaceDateOfAccountCre...	date	<input checked="" type="checkbox"/>
PlaceLike	int	<input checked="" type="checkbox"/>
PlaceDislike	int	<input checked="" type="checkbox"/>
PlaceProfilePicId	int	<input checked="" type="checkbox"/>

Fig 3.14: Database Table of Place Information.

### 3.5.3 Database Table for Tourist Information

Column Name	Data Type	Allow Nulls
TouristId	int	<input type="checkbox"/>
TouristName	varchar(MAX)	<input checked="" type="checkbox"/>
TouristEmail	varchar(MAX)	<input checked="" type="checkbox"/>
CountryId	int	<input checked="" type="checkbox"/>
CityId	int	<input type="checkbox"/>
TouristPhnNo	varchar(MAX)	<input checked="" type="checkbox"/>
TouristAddress	varchar(MAX)	<input checked="" type="checkbox"/>
TouristEmergencyPhnNo	varchar(MAX)	<input checked="" type="checkbox"/>
TouristsStatus	bit	<input checked="" type="checkbox"/>
TouristsHostStatus	bit	<input checked="" type="checkbox"/>
TouristAdminsPermit	bit	<input checked="" type="checkbox"/>
TouristDateOfAccountCr...	date	<input checked="" type="checkbox"/>
TouristPassword	varchar(MAX)	<input checked="" type="checkbox"/>
TouristProfilePicId	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Fig 3.15: Database Table of Tourist Information.

### 3.5.4 Database table for Host Information

Column Name	Data Type	Allow Nulls
HostId	int	<input type="checkbox"/>
TouristId	int	<input checked="" type="checkbox"/>
HostingCountryId	int	<input checked="" type="checkbox"/>
HostingCityId	int	<input checked="" type="checkbox"/>
HostingPlaceAddress	varchar(MAX)	<input checked="" type="checkbox"/>
Rent	varchar(MAX)	<input checked="" type="checkbox"/>
HostingStartTime	date	<input checked="" type="checkbox"/>
HostingStopTime	date	<input checked="" type="checkbox"/>
HostingDetail	nvarchar(MAX)	<input checked="" type="checkbox"/>
HostingStatus	bit	<input checked="" type="checkbox"/>
HostAdminsPermit	bit	<input checked="" type="checkbox"/>
HostDateOfAccountCrea...	date	<input checked="" type="checkbox"/>
HostLike	int	<input checked="" type="checkbox"/>
HostDislike	int	<input checked="" type="checkbox"/>
HostProfilePicId	int	<input checked="" type="checkbox"/>
HostTypeId	int	<input checked="" type="checkbox"/>
HostingPhnNo	varchar(MAX)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Fig 3.16: Database Table for Host Information.

### 3.5.5 Database Table for Vendor Details

Column Name	Data Type	Allow Nulls
VendorId	int	<input type="checkbox"/>
VendorName	varchar(MAX)	<input checked="" type="checkbox"/>
VendorTypeId	int	<input checked="" type="checkbox"/>
VendorEmail	varchar(MAX)	<input checked="" type="checkbox"/>
VendorPhnNo	varchar(MAX)	<input checked="" type="checkbox"/>
VendorAddress	varchar(MAX)	<input checked="" type="checkbox"/>
VendorOfficeName	varchar(MAX)	<input checked="" type="checkbox"/>
CountryId	int	<input checked="" type="checkbox"/>
CityId	int	<input checked="" type="checkbox"/>
VendorOfficePhnNo	varchar(MAX)	<input checked="" type="checkbox"/>
VendorOfficeAddress	varchar(MAX)	<input checked="" type="checkbox"/>
VendorOfficeWebsite	varchar(MAX)	<input checked="" type="checkbox"/>
VendorsVendorShipDetail	nvarchar(MAX)	<input checked="" type="checkbox"/>
VendorsStatus	bit	<input checked="" type="checkbox"/>
VendorOfferStatus	bit	<input checked="" type="checkbox"/>
VendorAdminsPermit	bit	<input checked="" type="checkbox"/>
VendorDateOfAccountCr...	date	<input checked="" type="checkbox"/>
VendorPassword	varchar(MAX)	<input checked="" type="checkbox"/>
VendorLike	int	<input checked="" type="checkbox"/>
VendorDislike	int	<input checked="" type="checkbox"/>
VendorProfilePicId	int	<input checked="" type="checkbox"/>
VendorOfficeMail	varchar(MAX)	<input checked="" type="checkbox"/>

Fig 3.17: Database Table for Vendor Details.

### 3.5.6 Database Tables for Country & City

Column Name	Data Type	Allow Nulls
CountryId	int	<input type="checkbox"/>
CountryName	varchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Fig 3.18: Table for Country.

Column Name	Data Type	Allow Nulls
CityId	int	<input type="checkbox"/>
CityName	varchar(50)	<input checked="" type="checkbox"/>
CountryId	int	<input checked="" type="checkbox"/>

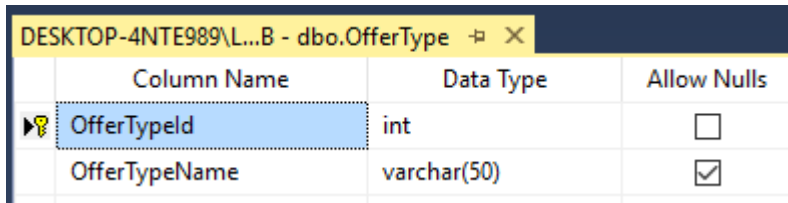
Fig 3.19: Table for City.

### 3.5.7 Database Table for Offer Information

Column Name	Data Type	Allow Nulls
OfferId	int	<input type="checkbox"/>
OfferName	varchar(MAX)	<input checked="" type="checkbox"/>
VendorId	int	<input checked="" type="checkbox"/>
CountryId	int	<input checked="" type="checkbox"/>
CityId	int	<input checked="" type="checkbox"/>
OfferTypeId	int	<input checked="" type="checkbox"/>
OfferPrice	varchar(MAX)	<input checked="" type="checkbox"/>
OfferDetail	nvarchar(MAX)	<input checked="" type="checkbox"/>
OfferStartTime	date	<input checked="" type="checkbox"/>
OfferStopTime	date	<input checked="" type="checkbox"/>
OfferStatus	bit	<input checked="" type="checkbox"/>
OfferAdminsPermit	bit	<input checked="" type="checkbox"/>
OfferDateOfAccountCre...	date	<input checked="" type="checkbox"/>
OfferLike	int	<input checked="" type="checkbox"/>
OfferDislike	int	<input checked="" type="checkbox"/>
OfferProfilePicId	int	<input checked="" type="checkbox"/>

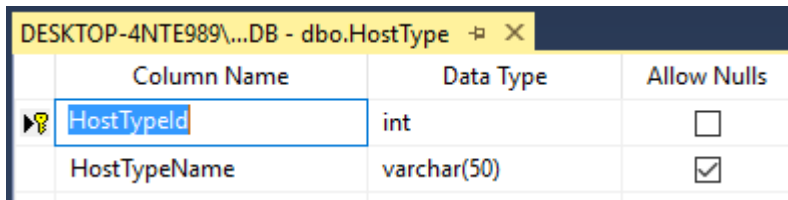
Fig 3.20: Table for Offer Information.

### 3.5.8 Database Table for Offer Type and Host Type



Column Name	Data Type	Allow Nulls
OfferTypeId	int	<input type="checkbox"/>
OfferTypeName	varchar(50)	<input checked="" type="checkbox"/>

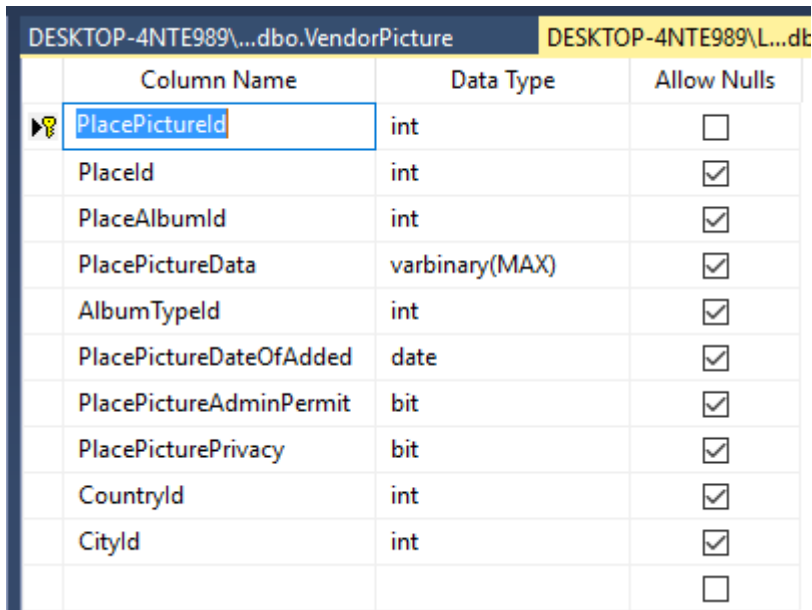
Fig 3.21: Table for Offer Type.



Column Name	Data Type	Allow Nulls
HostTypeId	int	<input type="checkbox"/>
HostTypeName	varchar(50)	<input checked="" type="checkbox"/>

Fig 3.22: Table for Host Type.

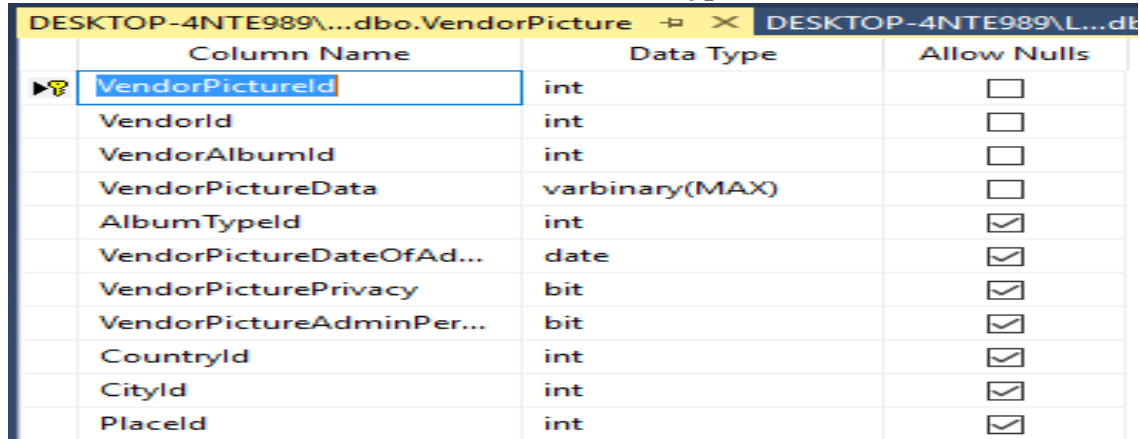
### 3.5.9 Database Table for Place Picture Data



Column Name	Data Type	Allow Nulls
PlacePictureId	int	<input type="checkbox"/>
PlaceId	int	<input checked="" type="checkbox"/>
PlaceAlbumId	int	<input checked="" type="checkbox"/>
PlacePictureData	varbinary(MAX)	<input checked="" type="checkbox"/>
AlbumTypeId	int	<input checked="" type="checkbox"/>
PlacePictureDateOfAdded	date	<input checked="" type="checkbox"/>
PlacePictureAdminPermit	bit	<input checked="" type="checkbox"/>
PlacePicturePrivacy	bit	<input checked="" type="checkbox"/>
CountryId	int	<input checked="" type="checkbox"/>
CityId	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

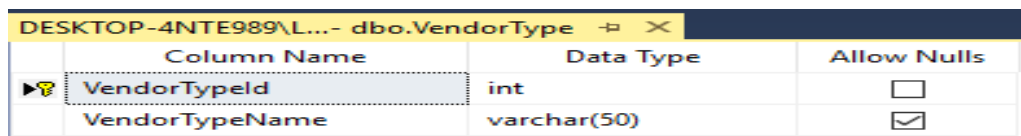
Fig 3.23: Table for Place Pictures.

### 3.5.10 Database Table for Vendor Picture and Vendor Type



Column Name	Data Type	Allow Nulls
VendorPictureId	int	<input type="checkbox"/>
VendorId	int	<input type="checkbox"/>
VendorAlbumId	int	<input type="checkbox"/>
VendorPictureData	varbinary(MAX)	<input type="checkbox"/>
AlbumTypeId	int	<input checked="" type="checkbox"/>
VendorPictureDateOfAd...	date	<input checked="" type="checkbox"/>
VendorPicturePrivacy	bit	<input checked="" type="checkbox"/>
VendorPictureAdminPer...	bit	<input checked="" type="checkbox"/>
CountryId	int	<input checked="" type="checkbox"/>
CityId	int	<input checked="" type="checkbox"/>
PlaceId	int	<input checked="" type="checkbox"/>

Fig 3.24: Table for Vendor Picture.



Column Name	Data Type	Allow Nulls
VendorTypeId	int	<input type="checkbox"/>
VendorTypeName	varchar(50)	<input checked="" type="checkbox"/>

Fig 3.25: Table for Vendor Type.

## 3.6 Software requirements

For creating the website, we mainly required the following software, installed on the hosting system,

### 3.6.1 Web Server

- Internet Information Services (IIS Express)

### 3.6.2 Database Language

- SQL

### 3.6.3 Server Side Language

- ASP.NET (C#)

### 3.6.4 Client Side Languages

- HTML, CSS, JavaScript

### 3.6.5 Platform

- Entity Framework 6.0 in MVC 5 methodology

## **3.7 Software Platform**

In this platform the tools are used fully open sources. For designing this project this HTML, CSS are used and in some cases JavaScript is used. IIS Express is used as web server and ASP.Net is used for database connecting codes SQL server is the database system.

### **3.7.1 HTML**

HTML (Hypertext Markup Language) is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page. The markup tells the Web browser how to display a Web page's words and images for the user. Each individual markup code is referred to as an element (but many people also refer to it as a tag). Some elements come in pairs that indicate when some display effect is to begin and when it is to end.

### **3.7.2 CSS**

A cascading style sheet (CSS) is a Web page derived from multiple sources with a defined order of precedence where the definitions of any style element conflict. The Cascading Style Sheet, level 1 recommendation from the World Wide Web Consortium ([W3C](http://www.w3.org)), which is implemented in the latest versions of the Netscape and Microsoft Web browsers, specifies the possible style sheets or statements that may determine how a given element is presented in a Web page. CSS gives more control over the appearance of a Web page to the page creator than to the browser designer or the viewer.

### **3.7.3 JavaScript**

JavaScript (JS) is a scripting languages, primarily used on the Web. It is used to enhance HTML pages and is commonly found embedded in HTML code. JavaScript is an interpreted language. Thus, it doesn't need to be compiled. JavaScript renders web pages in an interactive and dynamic fashion. This allowing the pages to react to events, exhibit special effects, accept variable text, validate data, create cookies, detect a user's browser, etc.

### **3.7.4 IIS Express**

Internet Information Services (IIS) is an extensible web server created by Microsoft for use with the Windows NT family. IIS Express is a solid, lightweight alternative that's self-contained and designed to make it easy for developers to build and test websites. It also offers the power of IIS 7. The server is integrated into WebMatrix and does not ship with IIS Manager for IIS 7. Instead, it is managed from within WebMatrix or from the icon. The Microsoft WebMatrix is an integrated suite of tools to manage ASP.NET and PHP development. It works with both Visual Studio and Visual Web Developer. it does not require

an administrator account nor any code changes. In fact, people can use it with all forms of ASP.NET applications. Moreover, people can build and test applications using it in Visual Studio, but it needs to be installed separately and manually launched from the command line.

### **3.7.5 ASP.NET**

ASP.NET is a Microsoft server-side Web technology. ASP.NET takes an object-oriented programming approach to Web page execution. Every element in an ASP.NET page is treated as an object and run on the server. An ASP.NET page gets compiled into an intermediate language by a .NET Common Language Runtime-compliant compiler. Then a JIT compiler turns the intermediate code to native machine code, and that machine code is eventually run on the processor. Because the code is run straight from the processor, pages load much faster than classic ASP pages, where embedded VBScript or JScript had to be continuously interpreted and cached. ASP.NET is used to create Web pages and Web services and is an integral part of Microsoft's .NET vision.

### **3.7.6 C# (C Sharp)**

C# is a hybrid of C and C++, it is a Microsoft programming language developed to compete with Sun's Java language. C# is an object-oriented programming language used with XML-based Web services on the .NET platform and designed for improving productivity in the development of Web applications.<sup>[1]</sup> C# boasts type-safety, garbage collection, simplified type declarations, versioning and scalability support, and other features that make developing solutions faster and easier, especially for COM+ and Web services. Microsoft critics have pointed to the similarities between C# and Java. ASP.NET is the portion of the .NET library used for making web sites where C# can be used to handle server side requirements.

### **3.7.7 Entity Framework and MVC**

Entity Framework is an Object Relational Mapper (ORM) which is a type of tool that simplifies mapping between objects in our software to the tables and columns of a relational database. It is an open source ORM framework for ADO.NET which is a part of .NET Framework. An ORM takes care of creating database connections and executing commands, as well as taking query results and automatically materializing those results as the application objects. An ORM also helps to keep track of changes to those objects, and when instructed, it will also persist those changes back to the database. It can generate the necessary database

commands for reading or writing data in the database and execute them. Queries can be expressed against domain objects using LINQ to entities. Entity Framework will execute the relevant query in the database and then materialize results into instances of domain objects.

The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects.



# 4 Evaluation of the System

## 4.1 User Interface

UI is a surface forming a common boundary between adjacent regions, bodies, substances or phases. A point at which independent systems or diverse group interact. A system of interaction or communication between a computer and another entity such as printer, or other computer. [7] So here is the UI of the Travelous. Where different type of options given to a user so that they can surf about tourism queries.

### 4.1.1 Home Page

The Home page for the general users

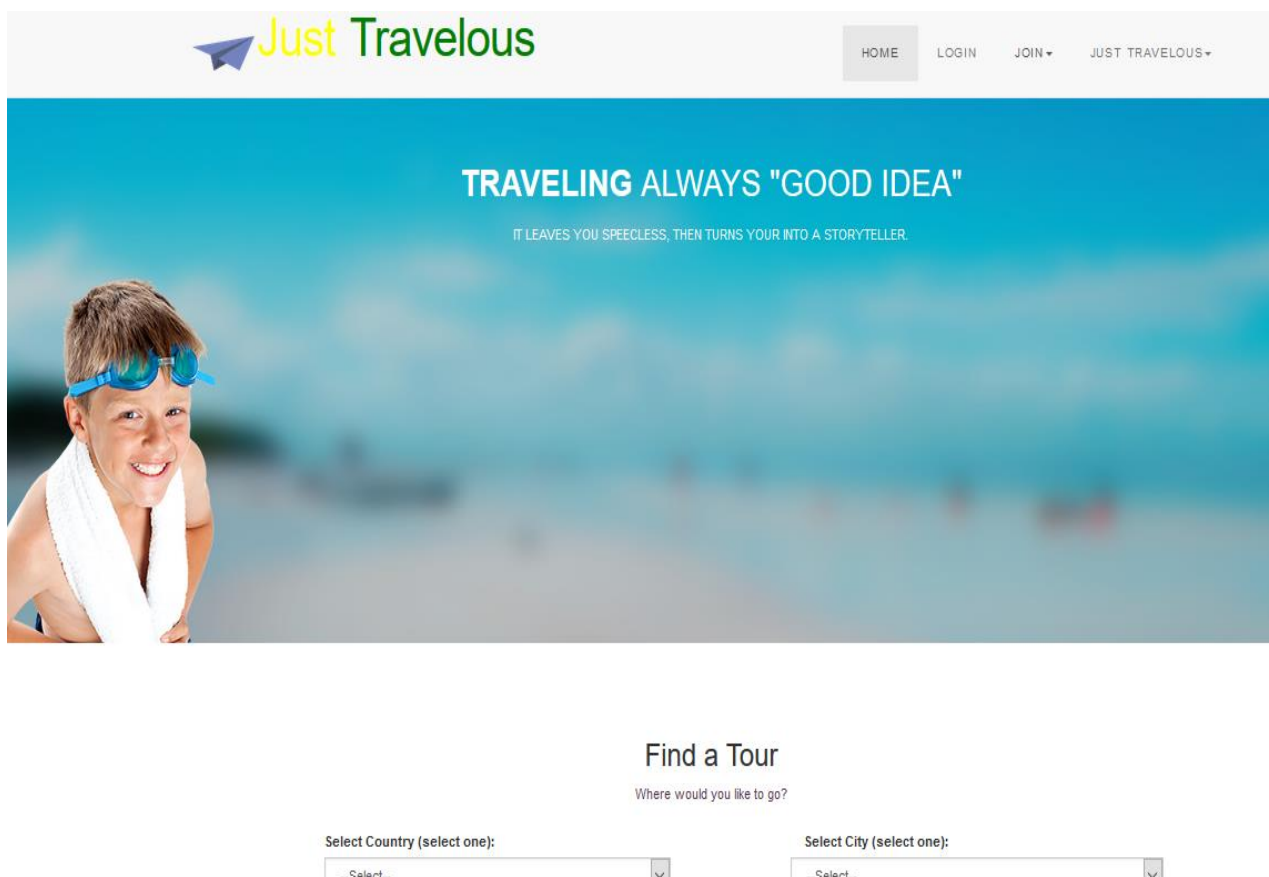
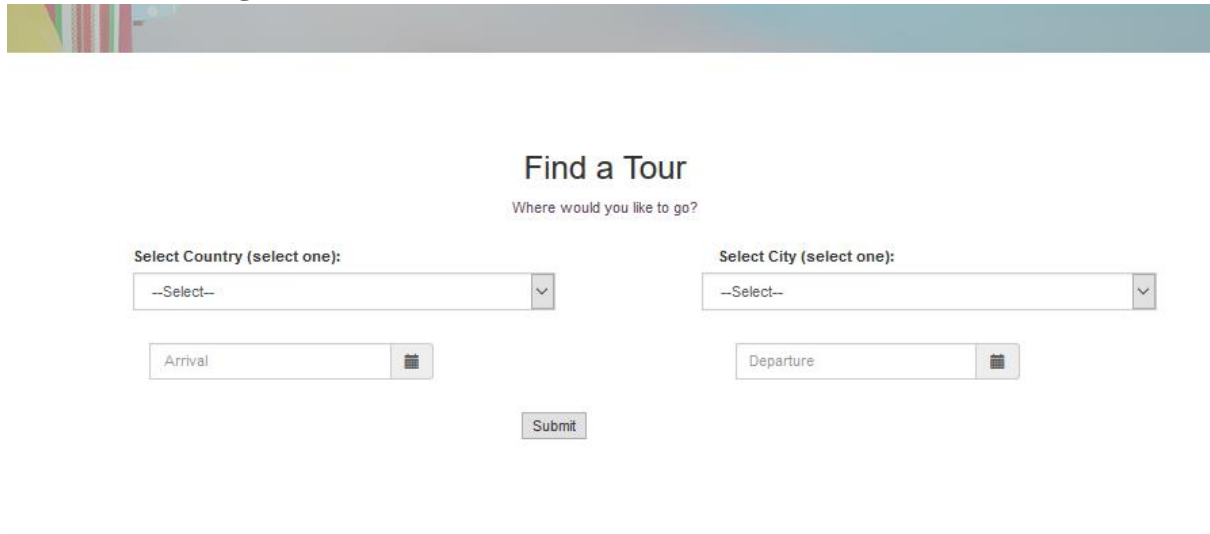


Fig 4.1: Home Page.

Description: In Home page, anyone can get access, log in or sign up as a registered tourist or vendor by going in the given link option.

## 4.1.2 Search Segment



The screenshot shows a search form titled "Find a Tour" with the subtitle "Where would you like to go?". It contains two dropdown menus: "Select Country (select one):" and "Select City (select one):", both currently showing "--Select--". Below these are two date pickers labeled "Arrival" and "Departure". A "Submit" button is centered at the bottom of the form.

Fig 4.2: Search Form.

Description: In the lower portion of the home page there is a segment named Find a Tour for general users, where they can choose Country, City, Arrival Date, Departure Date and search for any travel spot.

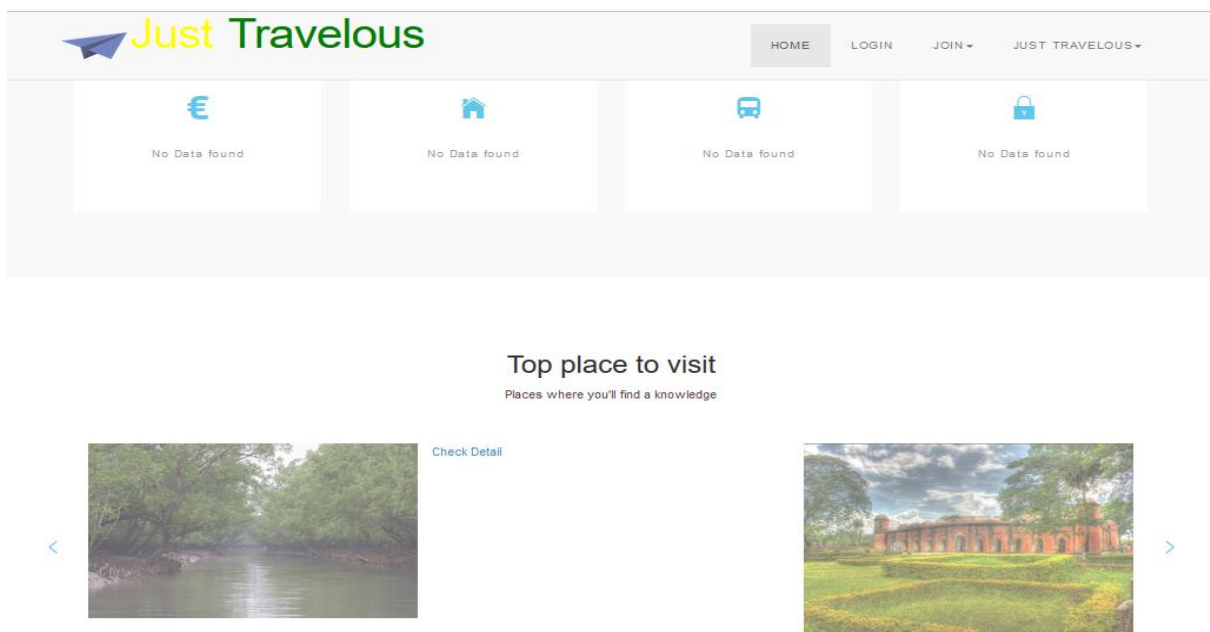


Fig 4.3: Results after searching a Travel spot.

### 4.1.3 Admin Sign In

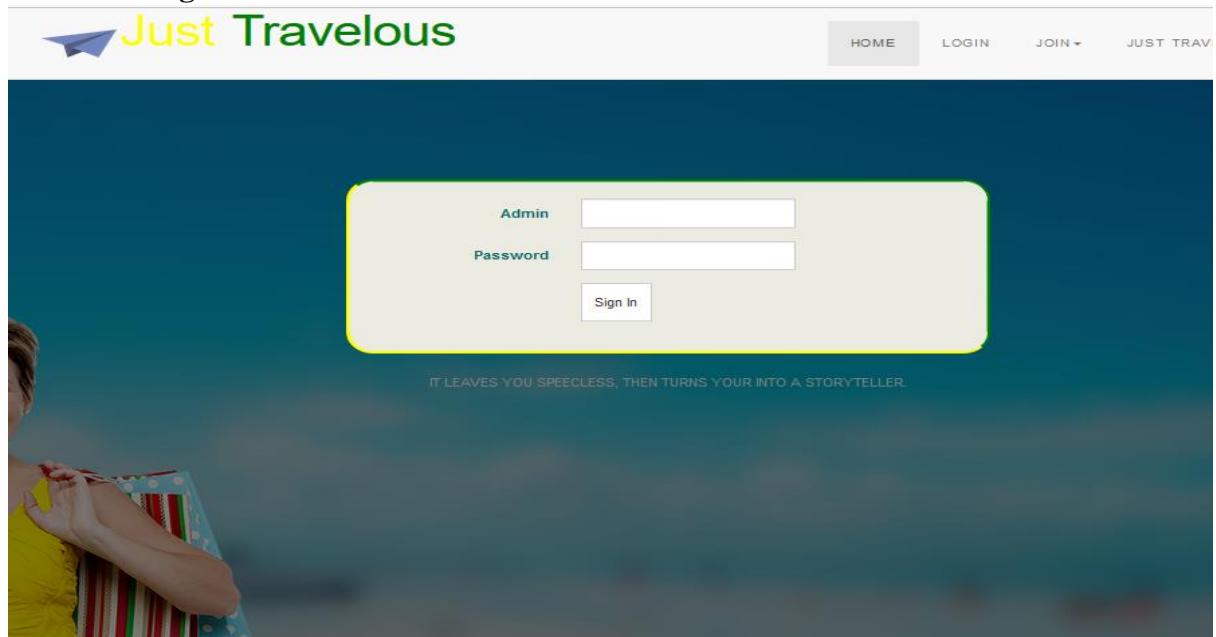


Fig 4.4: As admin Log In page.

Description: As admin has a unique name and password so he/she can only log in by that and get access to admin task by that.

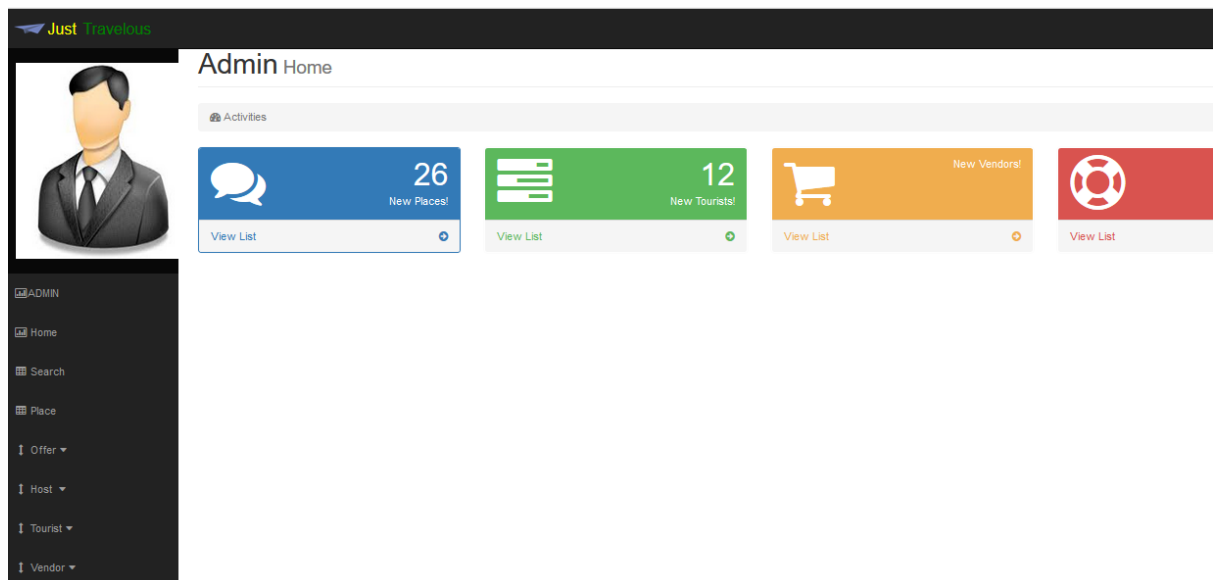


Fig 4.5: Admin Panel page after logged in.

#### 4.1.4 Vendor Sign Up Page

The screenshot shows the 'Just Travelous' website's vendor sign-up page. The page has a dark blue header with the logo on the left and navigation links (HOME, LOGIN, JOIN, JUST TRAVELOUS) on the right. The main content area is a light-colored box with a yellow border, containing a sign-up form. The form fields are as follows:

Field Label	Value
Company Name	asd
Company Type	Hotel
Country	Bangladesh
City	Dhaka
Company's Website	www.panpacific.com
Company's Email	s@gmail.com
Your Email	s@mail.com
Phone No.	+880255028008
Password	...
Confirm Password	...

At the bottom of the form is a button labeled 'Sign Up As a Vendor'.

Fig 4.6: Vendor Sign Up Page.

Description: A vendor has to give all these Information for his/her business purpose and security also. After signing up a vendor goes directly to the vendor panel page to do his/her business.

#### 4.1.5 Tourist Sign Up Page

The image shows a web page for signing up as a tourist. The header includes the logo 'Just Travelous' and navigation links 'HOME', 'LOGIN', and 'JOIN'. The sign-up form contains the following fields:

Field	Value
Name	asd
Country	Bangladesh
City	Rangpur
Email	s@mail.com
Phone No.	+02xxxxxxx
Password	•••
Confirm Password	•••

At the bottom of the form is a button labeled 'Sign Up As a Tourist'.

Fig 4.7: Tourist Sign Up page.

Description: By giving all necessary information a person can be a registered tourist of the site. Only after signing up a tourist can be a host or can add a new place.

#### 4.1.6 Add a New Place

The screenshot shows a web form titled "Add Place" with the following fields and components:

- Place Profile Picture:** A "Browse..." button and the text "No file selected."
- Name:** A text input field containing "Shundorban".
- Place Type:** A dropdown menu with "Natural Beauty Area" selected.
- Place Address:** A text input field containing "gerhat district in southern Ba".
- Country:** A dropdown menu with "Bangladesh" selected.
- City:** A dropdown menu with "Khulina" selected.
- Place Detail:** A rich text editor with a menu bar (File, Edit, View, Format) and a toolbar (undo, redo, bold, italic, bulleted list, numbered list, link, unlink). The text area contains "Mangrove Forest" and a small "Powered by TinyMCE" watermark.
- Update:** A button at the bottom of the form.

Fig 4.8: Form for adding a new Place.

Description: A registered user can a new place by giving its necessary Information. If the place is valid then the Admin will keep the place as open or she/she will block it after a certain time.

#### 4.1.7 Add a new Offer for Hosting

Offer Profile Picture  No file selected.

Host Type

CountryId

CityId

Business Phone No.

Price.

Address

Opening date.

Ending date.

Facilities/ Detail

File Edit View Format

Formats B I

Powered by TinyMCE

Fig 4.9: Form for adding a new hosting offer.

Description: For adding a new hosting offer a tourist has to give all these Information. Unless the offer will not be valid and The admin can block it any time.

#### 4.1.8 Host List Page

Just Travelous

HOME SEARCH SPECIFICALLY ADD PLACE SAMIA BARI JUST TRAVELOUS

#	Host	Phn No	Likes	Detail
1	Samia Bari	[REDACTED]	1	vbnjhhnmg <a href="#">Check Detail</a>

Fig 4.10: Host List after searching a place.

Description: From the basic search page (Home page) user can get a host list of the searched place.

# 5 Conclusion

## 5.1 Conclusion

Travelous (Information portal for tourist) was a new learning for me. By working on it I came to learn many things about new technologies like Entity Framework 6.0 and mvc5. Though these methods are very popular in outer countries but in our country these are not very popular. By creating this website using these technologies I have learned a lot methods of object oriented programming also. Working on this topic gave me a knowledge of thinking about a problem deeply and how can I implement it on my code easy. I wanted to work on this topic because nowadays Information portal for tourists has become a new experience and has greatly impacted the lives of Travelers and Vendors in its short time of existence. <sup>[7]</sup> It is expected to grow constantly in years to come with advancements in technology. Tourist portals has made travelers more active and efficient in their travels and also Vendor's product more effective for travelers. It has driven the travelling to a new level. It is forcing the vendors to make the necessary adjustments and changes to reach the new market of knowledgeable travelers. Online Tourist portals have evolved into sophisticated marketing channels for hoteliers and other vendors of all sizes. Never before have travelers been able to easily compare different travel options side-by-side in terms of price, time, and location. One of the greatest success stories in eCommerce is the development of Business to Business and Business to Consumer online sales of air travel, ground transportation, hotel accommodations, and tours. <sup>[1]</sup> In the past 10 years growing Internet traffic and a major industry policy shift created an economic environment that helped accelerate travel related businesses such as hotels and airlines to establish methods for direct online bookings. According to the American Society of Travel Agents, after decades of offering travel agents a standard commission of at least 10 % with no cap, many of the larger airline carriers in the United States stopped paying base commissions to travel agents, beginning in March 2002. Not surprisingly, some of the very companies that supplied traditional brick and mortar travel agencies with information and services became their chief competitors. Earlier people has to go to the hotels or to the other vendors in person or call them for information. They never got an idea of a place before going there. People has to struggle a lot for accommodation, food



and visiting a new place. But these online tourist portals offer access to markets once unattainable by small hoteliers and hostel owners. This offers consumers/travelers a versatile experience. Not only can they show the multitude of travel options, but also offer trusted reviews and accurate local information. Some online travel portals have created their own brands that consumers trust and recognize as thought-leaders. Hoteliers leverage these brand when they list their property with them. Travelers now can surf about a place, make bookings, check essential needs before going to a place. Even they can use these websites as their tour guide as some of these websites shows the direction of the place. Basically tourist portals are offering a traveler a package by giving all necessary information about traveling.

## **5.2 Future Work**

The Future plan of this project is to design, implementation and documentation in such a way that everyone can use this project for better performance. I will develop the site more dynamically and the database work as well. There are limitations in my project. In future I will add the following module for better improvement of the project:

- Online account verification
- Online booking system
- Online map director
- Online sms service for site holder.

# References

[1] gulfnews.com, 'Newsletter for tourism content' [2017] [online]

Available: <http://gulfnews.com/guides/travel/8-top-travel-and-tourism-websites-1.1857760> [Accessed date: 01.08.2017]

[2] Wikipedia.com, 'A online content website' [2017] [online]

Available: <https://en.wikipedia.org> [Accessed date: 01.08.2017]

[3] kayak.com 'A travel portal' [2017] [online]

Available: [www.kayak.co.in](http://www.kayak.co.in) [Accessed date: 01.08.2017]

[4] tripadvisor.com, 'A travel portal' [2017] [online]

Available: <https://tripadvisor.mediaroom.com/> [Accessed date: 02.08.2017]

[5] makemytrip.com 'A travel portal' [2017] [online]

Available: <https://www.makemytrip.com/> [Accessed date: 02.08.2017]

[6] trivago.com, 'A travel portal' [2017] [online]

Available: <https://www.trivago.com/> [Accessed date: 02.08.2017]

[7] technopedia.com, 'An educational website' [2017] [online]

Available: <https://www.technopedia.com/definition> [Accessed date: 02.08.2017]

# Appendix

Code:

## Admin Controller

```
public class AdminController : Controller
{
    public ActionResult SignIn(admin)
    {
        if (ModelState.IsValid)
        {
            try
            {
                bool result = adminClass.SignInAdmin(admin.AdminName, admin.AdminPassword);
                if (result == true)
                {
                    return RedirectToAction("UserPanel", "Home");
                }
            }
            else
            {
                ViewData["message"] = "Wrong Name or Password";
                return View(admin);
            }
        }
        catch (Exception ex)
        {
            return View("Error", new HandleErrorInfo(ex, "Admin", "SignIn"));
        }
        else
        {
            return View(admin);
        }
    }
}
```

## Home Controller

```
public class HomeController : Controller
{
    public ActionResult SignInUser(SignInUserViewModel siuvm)
    {
        if(ModelState.IsValid)
        {
            try
            {
```

```

if(siuvm.UserType==1)
{
return RedirectToAction("SignIn","Vendor",new {Email=siuvm.Email,
Password=siuvm.Password});
}
else if(siuvm.UserType == 2)
{
return RedirectToAction("SignIn", "Tourist", new { Email = siuvm.Email, Password =
siuvm.Password });
}
else
{
TempData["Problem"] = "You have to choose a user type option.";
return View();
}
}
catch (Exception ex)
{
return View("Error", new HandleErrorInfo(ex, "Home", "SignInUser"));
}
}else
{
return View(siuvm);
}
}
public ActionResult LogOut()
{
Session.Abandon();
Response.Cookies.Add(new HttpCookie("ASP.NET_SessionId", ""));
return RedirectToAction("Index");
}
}

```

## Host Controller

```
public ActionResult ShowHost(int HostId,int CountryId, int CityId, int Page)
{
    HostDetailSearchViewModel sp = new HostDetailSearchViewModel();
    if (HostId != 0)
    {
        var i = (from c in entity.Hosts
        join e in entity.HostPictures on c.HostProfilePicId
        equals e.HostPictureId into ppl
        from e in ppl.DefaultIfEmpty()
        join d in entity.Tourists on c.TouristId equals d.TouristId
        join f in entity.Countries on c.HostingCountryId equals f.CountryId
        join g in entity.Cities on c.HostingCityId equals g.CityId
        join k in entity.HostTypes on c.HostTypeId equals k.HostTypeId
        where c.HostId==HostId
        orderby c.HostDateOfAccountCreation descending
        select new
        {
            c.HostId,
            d.TouristName,
            e.HostPictureData,
            d.TouristId,
            f.CountryId,
            g.CityId,
            k.HostTypeId,
            c.HostingPhnNo,
            c.HostingPlaceAddress,
            f.CountryName,
            g.CityName,
            c.HostingStartTime,
            c.HostingStopTime,
```

```

c.HostingDetail,
c.Rent,
k.HostTypeName,
c.HostingStatus,
c.HostAdminsPermit
}).First();
if (i!=null)
{
sp.HostPictureData = i.HostPictureData;
sp.HostId = i.HostId;
sp.TouristId = Convert.ToInt32(i.TouristId);
sp.HostName = i.TouristName;
sp.HostAddress = i.HostingPlaceAddress;
sp.HostTypeName = i.HostTypeName;
sp.HostPhnNo = i.HostingPhnNo;
sp.CityId = i.CityId;
sp.CountryId = i.CountryId;
sp.HostTypeId = i.HostTypeId;
sp.HostPrice = i.Rent;
sp.HostingStartTime = Convert.ToDateTime(i.HostingStartTime);
sp.HostingEndTime = Convert.ToDateTime(i.HostingStopTime);
string st = CutDetail.StripHTML(i.HostingDetail);
sp.HostShortDetail = CutDetail.Truncate(st, 40);
sp.HostFullDetail = i.HostingDetail;
sp.HostTypeName = i.HostTypeName;
sp.hostAdminPermit = Convert.ToBoolean(i.HostAdminsPermit);
sp.HostStatus = Convert.ToBoolean(i.HostingStatus);
}
var q = sp;
TempData["CountryId"] = CountryId;
TempData["CityId"] = CityId;

```

```

return View(q);
}
else
{
return RedirectToAction("Search", "Home", new { CountryId = CountryId, CityId = CityId,
Arrival = "", Departure = "" });
}
}

```

### Vendor Controller

```

Public ActionResult SignUp()
{
Vendor vendor = new Vendor();
vendor.VendorTypeId = VendorTypeId;
vendor.VendorEmail = vsuvm.VendorEmail;
vendor.VendorPhnNo = vsuvm.VendorPhnNo;
vendor.VendorOfficeName = vsuvm.VendorOfficeName;
vendor.CountryId = CountryId;
vendor.CityId = CityId;
if (vsuvm.VendorOfficeWebsite==null)
{
vendor.VendorOfficeWebsite = "Not given yet";
}
else
{
vendor.VendorOfficeWebsite = vsuvm.VendorOfficeWebsite;
}
vendor.VendorName = "Not given yet";
vendor.VendorAddress = "Not given yet";
vendor.VendorOfficeMail = vsuvm.VendorOfficeMail;
vendor.VendorOfficePhnNo = "Not given yet";
vendor.VendorsVendorShipDetail= "Not given yet";

```

```

vendor.VendorOfficeAddress= "Not given yet";
vendor.VendorsStatus = true;
vendor.VendorOfferStatus = false;
vendor.VendorAdminsPermit = true;
vendor.VendorDateOfAccountCreation = DateTime.Now;
vendor.VendorPassword = vsuvm.VendorPassword;
vendor.VendorFavourite = 0;
entity.Vendors.Add(vendor);
if (entity.SaveChanges() > 0)
{
string userCode = "vendor";
Session["userCode"] = userCode.ToString();
Session["VendorEmail"] = vsuvm.VendorEmail;
Session["VendorOfficeName"] = vsuvm.VendorOfficeName;
var d = (from c in entity.Vendors where c.VendorEmail == vsuvm.VendorEmail select
c).FirstOrDefault();
int vendorId = Convert.ToInt32(d.VendorId);
Session["VendorId"] = vendorId;
Session["CountryId"] = vsuvm.CountryId;
Session["CityId"] = vsuvm.CityId;
Nullable<int> ProPicId = d.VendorProfilePicId;
if (ProPicId != null)
byte[] pic = (from c in entity.VendorPictures where c.VendorId == vendorId &&
c.AlbumTypeId == 1 orderby c.VendorPictureId descending select
c.VendorPictureData).First();
if (pic != null)
{
Session["pp"] = pic;
}
}
return RedirectToAction("UserPanel", "Home");
}

```



## Tourist Controller

```
Public ActionResult SignIn()
{
    bool result = tl.SignInLogic(Email, Password);
    if (result == true)
    {
        int touristId = (int)Session["TouristId"];
        var d = (from c in entity.Tourists where c.TouristId == touristId select c).FirstOrDefault();
        Nullable<int> ProPicId = d.TouristProfilePicId;
        if (ProPicId != null)
        {
            byte[] pic = (from c in entity.TouristPictures where c.TouristId == touristId &&
                c.AlbumTypeId == 1 orderby c.TouristPictureId descending select
                c.TouristPictureData).First();
            if (pic != null)
            {
                Session["pp"] = pic;
            }
        }
        return RedirectToAction("Index", "Home");
    }
    else
    {
        Session["message"] = "no";
        return RedirectToAction("SignInUser", "Home");
    }
}
```

## Offer Controller

```
public ActionResult AddOffer()
{
    if ((string)Session["userCode"] == "vendor")
    {
        int VendorId = (int)Session["VendorId"];
        var d = (from c in entity.Vendors where c.VendorId == VendorId select c).FirstOrDefault();
        AddOfferViewModel aovm = new AddOfferViewModel();
        aovm.CountryId = Convert.ToInt32(d.CountryId);
        aovm.CityId = Convert.ToInt32(d.CityId);
        aovm.VendorId = Convert.ToInt32(d.VendorId);
        aovm.OfferTypeList = new SelectList(entity.OfferTypes, "OfferTypeId", "OfferTypeName");
        return View(aovm);
    }
    else
    {
        Session["message"] = "please log in to add your offer.";
        return RedirectToAction("Index", "Home");
    }
}
```

## Place Controller

```
public ActionResult AddPlace(AddPlaceViewModel apvm, HttpPostedFileBase AddPicture)
{
    if (ModelState.IsValid)
    {
        try
        {
            int countryId = Convert.ToInt32(apvm.CountryId);
            int cityId = Convert.ToInt32(apvm.CityId);
```

```

bool result;
result = pl.CheckPlaceExist(apvm.PlaceName, countryId, cityId);
if (result == false)
{
int placeTypeId = Convert.ToInt32(apvm.PlaceTypeId);
Place place = new Place();
place.PlaceName = apvm.PlaceName;
place.PlaceTypeId = Convert.ToInt32(apvm.PlaceTypeId);
place.CountryId = Convert.ToInt32(apvm.CountryId);
place.CityId = Convert.ToInt32(apvm.CityId);
place.PlaceAddress = apvm.PlaceAddress;
if (apvm.PlaceDetail == null)
{
place.PlaceDetail = "Not Given yet.";
}
else
{
place.PlaceDetail = apvm.PlaceDetail;
}
place.PlaceAdminsPermit = true;
place.PlaceDateOfAccountCreation = DateTime.Now;
place.PlaceFavourite = 0;
PlacePicture pp = new PlacePicture();
entity.Places.Add(place);

if (entity.SaveChanges() > 0)
{

var p = (from c in entity.Places where c.PlaceName == apvm.PlaceName && c.CountryId ==
apvm.CountryId && c.CityId == apvm.CityId select new { c.PlaceId, c.CountryId, c.CityId
}).FirstOrDefault();
if (AddPicture != null)

```

```

{
PlaceAlbum pa = new PlaceAlbum();
pa.PlaceAlbumName = "Profile Picture";
pp.PlacePicturePrivacy = true;
pp.PlacePictureDateOfAdded = DateTime.Now;
entity.PlacePictures.Add(pp);
if (entity.SaveChanges() > 0)
{
Place d= (from c in entity.Places where c.PlaceId == p.PlaceId select c).FirstOrDefault();
int pictureId = (from c in entity.PlacePictures
select c.PlacePictureId).First();
d.PlaceProfilePicId = pictureId;
entity.SaveChanges();
}
}
ModelState.Clear();
TempData["Success"] = "Added Successfully";

}
else
{

TempData["Success"] = "Problem occured";
}

}
else
{

TempData["Success"] = "Duplicacy found";
}
}

```

```
AddPlaceViewModel apvm1 = new AddPlaceViewModel();
apvm1.PlaceTypeList = new SelectList(entity.PlaceTypes, "PlaceTypeId",
"PlaceTypeName");
return View(apvm1);
}
catch (Exception ex)
{
return View("Error", new HandleErrorInfo(ex, "Place", "AddPlace"));
}
}
}
}
```

----- END -----