**Contents**

1. Introduction
2. Database Planning
3. Requirements Analysis
   1. Scope
   2. Objectives
   3. Project Outcomes
   4. Analysis
4. Database Design
   1. ER Diagram
   2. Design Analysis
5. Database Implementation
6. Testing
7. Conclusion

**Introduction**

Nearly everyone goes on a vacation. A tourism management system would play a vital role in planning the perfect trip. The tourism management system allows the user of the system to access all the details such as packages, bookings, transportations etc. The main purpose is to help tourism companies to manage their customers easily and increase the trust of the users to the travel company as well. The proposed system maintains a centralized repository to make necessary travel arrangements and to retrieve information easily.

Once the bookings are confirmed all the tour details such as user details as well all the relevant details like costs, vehicle details, vehicle driver details, duration of the tour and every single other thing will be available to the user. All that a user need to do is click using the mouse and everything will be available to them.

**Database Planning**

The goal is to develop a complete tourism management system database. In this work, diﬀerent general or specific use cases for the problem has been considered, entities are thought of and related work flows have been established. At first, we thought about the project’s requirements. For example, what kind of information the database has to store and what are the scopes. We thought of the project’s outcome and objectives. We also thought about the problems of some tourism management systems that already exist so that we may avoid them altogether. Then we analyzed the requirements. For example, the hardware and software requirements to run the database. Finally, we chose a favorable language to implement the database.

**Requirements Analysis**

**Scope**

Tourism Management System is an integrated software developed for tour operating companies. The main aim of this project is to develop the back-end part of the software which is the database to help the tourism companies to manage their customers, transportations, users, packages, bookings and emergencies. It makes all operation of the tour company easy and accurate. The standalone platform makes tourism management easy by handling agencies requests and providing servers for the customers located at different parts of the various cities. Different modules have been incorporated in this project to handle different parts and sector of the tour management field.

**Objectives**

The tourism Management System is a web-based application that maintains a centralized repository of all related information. The objective of the project is to design a database of this system that automates the process and activities of a travel agency and customer details. The purpose is to design a system using which one can perform all operations related to travelling and sightseeing.

**Project Outcomes**

There are some objectives or outcomes of every project. Our project’s objectives are given below briefly:

* This database will provide to connect directly with customers and agents through the internet.
* The database will provide all the information about the inbound and outbound tour packages.
* The database will store all the tourist’s information.
* The database will store all the information about travelling locations.
* The database will provide a variety of packages.
* The database will store all the bookings information
* The database will provide all the information about transportation

**Analysis**

**Project Development Language:**

We are using MySQL language to create the database. We are using MySQL Workbench 8.0 to create our database project on the Tourism Management System.

**Hardware Requirements:**

* CPU: Intel Core or Xeon 3GHz (or Dual Core 2GHz) or equal AMD CPU
* Cores: Single (Dual/Quad Core is recommended)
* RAM: 4 GB (6 GB recommended)
* Graphic Accelerators: NVidia or ATI with support of OpenGL 1.5 or higher
* Display Resolution: 1280×1024 is recommended, 1024×768 is minimum

**Software Requirements:**

The following operating systems are officially supported:

* Windows 7 (64-bit, Professional level or higher)
* Mac OS X 10.6.1+
* Ubuntu 9.10 (64bit)
* Ubuntu 8.04 (32bit/64bit)

For convenience the following builds are also available:

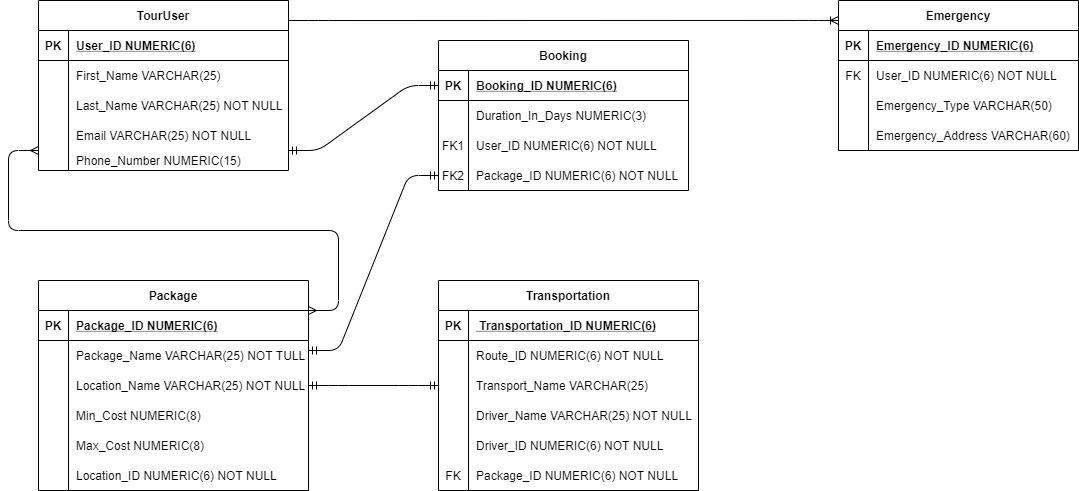
* Windows XP SP3, Vista
* Mac OSX (10.5 and 10.6) Intel
* Ubuntu 8.04 (i386/x64)
* Ubuntu 9.04 (i386/x64)
* Fedora 11 (i386/x64)

MySQL Workbench also has the following general requirements:

* The Microsoft .NET 3.5 Framework.
* Cairo 1.6.0 or later
* glib-2.10
* libxml-2.6
* libsigc++ 2.0
* pcre
* libzip

**Database Design**

**ER Diagram**



**Design Analysis**

Our database project has 5 tables which are TourUser, Bookings, Packages, Transportation and Emergency.

* *TourUser*: This table covers all the details about the tourists which they can register by themselves by adding their information in the system. This table contains First\_Name, Last\_Name, User\_ID, Email and Phone\_Number columns where User\_ID is the primary key.
* *Packages*: The database administrator can create packages by creating package details. The packages table contains Package\_ID, Package\_Name, Location\_Name, Min\_Cost, Max\_Cost, Location\_ID columns where Package\_ID is the primary key which also acts as a foreign key in the Booking table. A single tourist can book multiple packages and a single package consists of multiple tourists.
* *Bookings*: Bookings features enable the agents to get direct online bookings from their customers. The bookings table contains Booking\_ID, Duration\_In\_Days, User\_ID and Package\_ID columns where Booking\_ID is

the primary key, User\_ID and Package\_ID are the foreign keys referencing the TourUser and Packages tables respectively.

* *Transportation*: Transportation table assists the tourists and agents to move from one place to another where all the transportations details are available. It consists of Transportation\_ID, Route\_ID, Transport\_Name, Driver\_Name, Driver\_ID and Package\_ID columns where Transportation\_ID is the primary key and Package\_ID is the foreign key referencing the Packages table.
* *Emergency*: In case of any emergency such as health deterioration, loss of property, fire etc. the tourists can inform the system for help. Emergency table provides columns such as Emergency\_ID, User\_ID, Emergency\_Type, and Emergency\_Address. Where Emergency\_ID is the primary key and User\_ID is the foreign key referencing the TourUser table. A single tourist can have multiple emergencies.

**Database Implementation**

**MySQL codes for creating the tables:**

CREATE TABLE TourUser(

First\_Name VARCHAR(25),

Last\_Name VARCHAR(25) NOT NULL,

User\_ID NUMERIC(6) PRIMARY KEY,

Email VARCHAR(25) NOT NULL,

Phone\_Number NUMERIC(15));

CREATE TABLE Packages(

Package\_ID NUMERIC(6) PRIMARY KEY,

Package\_Name VARCHAR(25) NOT NULL,

Location\_Name VARCHAR(25) NOT NULL,

Min\_Cost NUMERIC(8),

Max\_Cost NUMERIC(8),

Location\_ID NUMERIC(6) NOT NULL);

CREATE TABLE Bookings(

Booking\_ID NUMERIC(6) PRIMARY KEY,

Duration\_In\_Days NUMERIC(3),

User\_ID NUMERIC(6) NOT NULL,

Package\_ID NUMERIC (6) NOT NULL,

FOREIGN KEY (User\_ID) REFERENCES TourUser(User\_ID),

FOREIGN KEY (Package\_ID) REFERENCES Packages(Package\_ID));

CREATE TABLE Transportation(

Transportation\_ID NUMERIC(6) PRIMARY KEY,

Route\_ID NUMERIC(6) NOT NULL,

Transport\_Name VARCHAR(25),

Driver\_Name VARCHAR(25) NOT NULL,

Driver\_ID NUMERIC(6) NOT NULL,

Package\_ID NUMERIC(6) NOT NULL,

FOREIGN KEY (Package\_ID) REFERENCES Packages(Package\_ID));

CREATE TABLE Emergency(

Emergency\_ID NUMERIC(6) PRIMARY KEY,

User\_ID NUMERIC(6) NOT NULL,

Emergency\_Type VARCHAR(50),

Emergency\_Address VARCHAR(60),

FOREIGN KEY (User\_ID) REFERENCES TourUser(User\_ID));

**MySQL codes for populating the tables:**

INSERT INTO TourUser VALUES

('Samin', 'Yasar', 101, 'syasar@gmail.com', 8801724345789), ('Sami', 'Alim', 102, 'salim@yahoo.com', 8801625643890), ('Mim', 'Sarwar', 103, 'msarwar@yahoo.com', 8801711789546),

('Jesan','Rana', 104, 'jrana@live.com', 8801616057392), ('Samia', 'Tasnim', 105, 'stasnim@gmail.com', 8801911209876), ('Robin', 'Roy', 106, 'rroy@yahoo.com', 8801824567865), ('Taha', 'Islam', 107, 'tislam@gmail.com', 8801674556876), ('Abrar', 'Hasan', 108, 'ahsan@gmail.com', 880172456792), ('Kashfia', 'Rahman', 109, 'krahman@live.com', 8801911436894), ('Badhon', 'Ahmed', 110, 'bahmed@gmail.com', 880162411765);

INSERT INTO Packages VALUES

(01, 'Beach Destination', 'Cox’s Bazar', 4500, 7900, 1100),

(02, 'Beach Destination', 'Saint Martin', 5500, 8800,1101),

(03, 'Jungle Safari', 'Sundarban', 8000, 11000, 1102),

(04, 'Heritage of Dhaka','Buriganga', 2000, 3500, 1103),

(05, 'Mountain Adventures','Rangamati', 3500, 5400, 1104),

(06, 'Mountain Adventures', 'Bandarban', 3600 ,5000, 1105),

(07, 'Mountain Adventures','Sajek valley',4200, 5600, 1106),

(08, 'Land of Tea Estates','Sylhet', 4800, 6000, 1107),

(09, 'Land of Tea Estates','Srimangal', 5200,6500, 1108),

(10, 'Hill tracts','Khagrachari', 6200,7900, 1109),

(11, 'Beach Explore', 'Kuakata', 5800,7800, 1110); INSERT INTO Bookings VALUES

(10, 3, 103, 02),

(20, 2, 101, 01),

(30, 1, 101, 04),

(40, 4, 109, 03),

(50, 5, 108, 02),

(60, 3, 106, 05),

(70, 4, 104, 07),

(80, 2, 104, 11),

(90, 1, 109, 04),

(100, 5, 107, 10);

INSERT INTO Transportation VALUES

(743956, 10001, 'Hanif', 'Rafiq', 983981, 09),

(112302, 10301, 'Sb', 'Karim', 342981, 10),

(132103, 10401, 'Greenline', 'Shupriyo', 982341, 11),

(103412, 13043, 'Hanif', 'Jamal', 653981, 07),

(103123, 10311, 'Syamoli', 'Hasan', 983561, 06),

(343106, 10004, 'Sb', 'Shafiq', 945681, 04),

(102317, 10303, 'Syamoli', 'Sakib', 984561, 05),

(101238, 14041, 'Greenline', 'Ronnie', 456981, 02),

(102139, 10451, 'Hanif', 'Abdul', 984541, 01),

(111230, 12101, 'Hanif', 'Pasha', 553981, 03);

INSERT INTO Emergency VALUES

(101, 109, 'Health issue', 'Azimpur road 1205'),

(102, 108, 'Loss of personal belonging', null),

(103, 101, 'Toxic gas releases', 'Bangla Bazar road Dhaka 1205'),

(104, 107, 'Explosions', 'Dokhin Bazar Kulaura Moulvibazar'),

(105, 101, 'Floods', 'Kolatoli Cox Bazar 4700'),

(106, 103, 'Loss of property/money', null),

(107, 105, 'Tropical storms', ' Zubli Road Booking Booth Enayet Bazar Ctg'),

(108, 102, 'Fire', null),

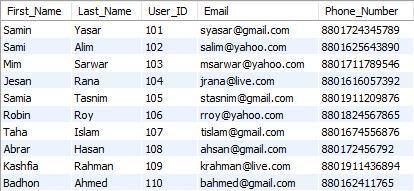
(109, 102, 'Wildfire emergency', 'Moulvibazar Road Srimangal'),

(110, 103, 'Criminal act', null);

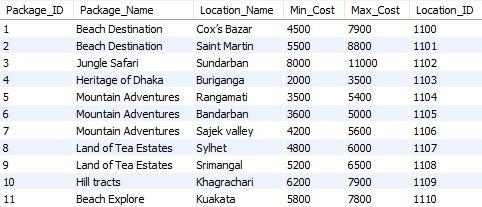
**Testing**

**Screenshots of all the tables from MySQL Workbench:**

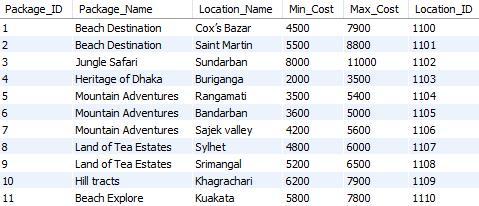
*TourUser*



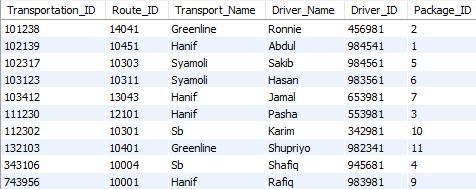
*Packages*



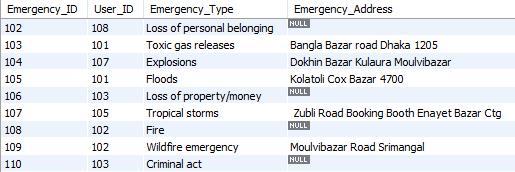
*Bookings*



*Transportation*



*Emergency*



**Some queries and their produced results are below to test the database:**

Creating a view joining all the tables and showing all the columns without repetition:

CREATE VIEW ALLTABLES

AS SELECT tu.User\_ID, tu.First\_Name, tu.Last\_Name, tu.Email,

tu.Phone\_Number, b.Booking\_ID, b.Duration\_days, p.Package\_ID,

p.Package\_Name, p.Location\_Name, p.Location\_ID, p.Max\_Cost,

p.Min\_Cost, t.Transportation\_ID, t.Transport\_Name, t.Route\_ID,

t.Driver\_Name, t.Driver\_ID, e.Emergency\_ID, e.Emergency\_Type,

e.Emergency\_Address

FROM touruser tu

LEFT OUTER JOIN bookings b

ON tu.User\_ID = b.User\_ID

LEFT OUTER JOIN packages p

ON p.Package\_ID = b.Package\_ID

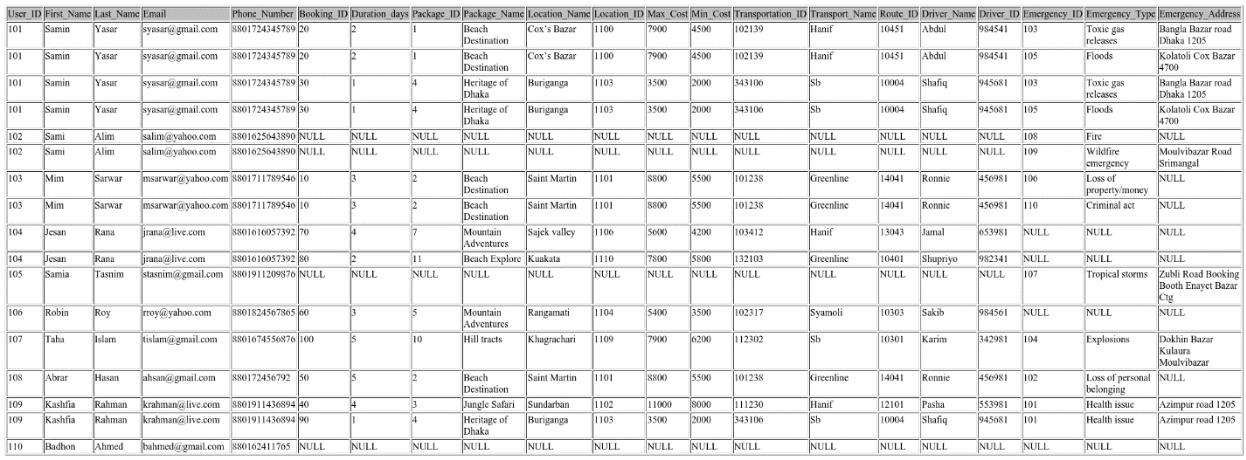
LEFT OUTER JOIN transportation t

ON t.Package\_ID = p.Package\_ID

LEFT OUTER JOIN emergency e

ON e.User\_ID = tu.User\_ID

ORDER BY User\_ID;



Finding all the tourists who booked for the location Sundarban:

SELECT tu.User\_ID, tu.First\_Name, tu.Last\_Name, b.Booking\_ID, b.Duration\_days, p.Package\_ID, p.Package\_Name, p.Location\_Name, p.Location\_ID, p.Max\_Cost, p.Min\_Cost FROM touruser tu

LEFT OUTER JOIN bookings b

ON tu.User\_ID = b.User\_ID

LEFT OUTER JOIN packages p

ON p.Package\_ID = b.Package\_ID

WHERE Location\_Name = 'Sundarban'

ORDER BY User\_ID;



Finding all the tourists who booked for the package Mountain Adventures:

SELECT tu.User\_ID, tu.First\_Name, tu.Last\_Name, b.Booking\_ID, b.Duration\_days, p.Package\_ID, p.Package\_Name, p.Location\_Name, p.Location\_ID, p.Max\_Cost, p.Min\_Cost FROM touruser tu

LEFT OUTER JOIN bookings b

ON tu.User\_ID = b.User\_ID

LEFT OUTER JOIN packages p

ON p.Package\_ID = b.Package\_ID

WHERE Package\_Name = 'Mountain Adventures' ORDER BY User\_ID;



Finding all the tourists who will be using the transportation named Hanif:

SELECT tu.User\_ID, tu.First\_Name, tu.Last\_Name, b.Booking\_ID,

p.Package\_ID, p.Package\_Name, p.Location\_Name, p.Location\_ID,

t.Transport\_Name

FROM touruser tu

LEFT OUTER JOIN bookings b

ON tu.User\_ID = b.User\_ID

LEFT OUTER JOIN packages p

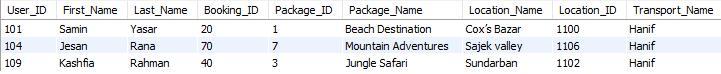
ON p.Package\_ID = b.Package\_ID

LEFT OUTER JOIN transportation t

ON t.Package\_ID = p.Package\_ID

WHERE Transport\_Name = 'Hanif'

ORDER BY User\_ID;



**Conclusion**

This project “Tourism Management System” can be used to book tours from anywhere by a single dynamic website which will help the tourists to know all about the places and tour details.

This database was successfully created and stored all the tourist’s details, tour packages, bookings, transportation information and emergencies tour details into the database. Normalization is applied on all the tables and are found to be in 3NF. The application was tested very well and the errors were properly debugged. Testing also concluded that the performance of the system is satisfactory. All the necessary output is generated. Working with such a system can enable the user to get any information with low performance cost and increased throughput. Further enhancements can be made to the project, so that the database functions in a very attractive and useful manner than the present one. It is concluded that the database works well and satisfies the needs of the tour companies and tour users as well.