# Functional Specification, E-R Model and Test Plan for Database and Information Systems Project

October 14, 2013

## 1 Introduction

This document describes the functional specifications, the E-R model and the Test plan of the application being developed to meet the requirements of the completion of the Database and Information Systems Lab project. The project details are as below:

- Name of the Project : *Trawell*
- Team Members : Mridul Garg (110050030), Anand Soni (110050037), Sanchit Garg (110050035), Himanshu Roy (110050019)

## 2 Project Domain

- Tourism. The application will serve as a travel planner and guide for tourists.

## 3 Application Specifications

Based on the category of the user, the application is supposed to provide the following functionalities:

- For an unregistered user : An unregistered user is any random visitor who comes across and explores the application.
  - Search for tourist places by country, state or city
  - Search for a tourist place by its name
- For a registered user : A registered user is one who has filled in her personal details and has been provided login credentials.
  - Search for tourist places by country, state or city
  - Search for a tourist place by its name
  - Search for hotels by city or tourist spot
  - Search for speciality cuisines of the region
  - Rate places visited and hotels stayed in
  - See the places and hotels that she has rated
  - Maintain a wishlist for the places she wants to visit

  - Prepare and modify a travel plan for her tour
  - Maintain a history which records all the details of her trip
  - Delete history items
  - Maintain a manual schedule as against the plan
  - Update personal information
- For Administrators : An administrator is the owner of this application who can create and modify the database and can also add more users to it (employees or customers).
  - Modify the tourism database
  - Create logins for recruited employees
- For employees : An employee (not necessarily a user) has the power to modify the database.
  - Modify the tourism database

## 4 Aspects Modelled

As a part of this application, we have created 10 entities and modelled them as below:

- Modelled Customers, Administrator, Countries, States, Cities, Tourist spots, Hotels, Schedule.
- Modelled travel plan, user's tour history.

## 5 Aspects Not Modelled

The following aspects have not been modelled in the application:

- Review system, Festivals, Shopping, Entertainment and other activities.

## 6 Role of Each Member

This section roughly describes the parts of the project that each member will contribute to. However, the actual contributions may differ.

- Mridul Garg
  - Design of the Functional Specification
  - Design of relational database and screen
  - Normalization and design of final schema
  - Writing SQL scripts for schema and other programming tasks.
- Anand Soni
  - Design of the Functional Specification
  - E/R modelling and creation of test plan
  - Design of relational database and screen

  - Writing SQL scripts for schema and other programming tasks.

- Sanchit Garg
  - Design of the Functional Specification
  - E/R modelling and creation of test plan
  - Normalization and design of final schema
  - Writing SQL scripts for schema and other programming tasks.

- Himanshu Roy
  - Design of the Functional Specification
  - E/R modelling and creation of test plan
  - Normalization and design of final schema
  - Writing SQL scripts for schema and other programming tasks.

## 7 Possible Value Addition

The following features might prove to be further value additions to this project which we have not planned to model/implement:

- Hotel Bookings
- Travel Bookings and planning

## 8 The E-R Model

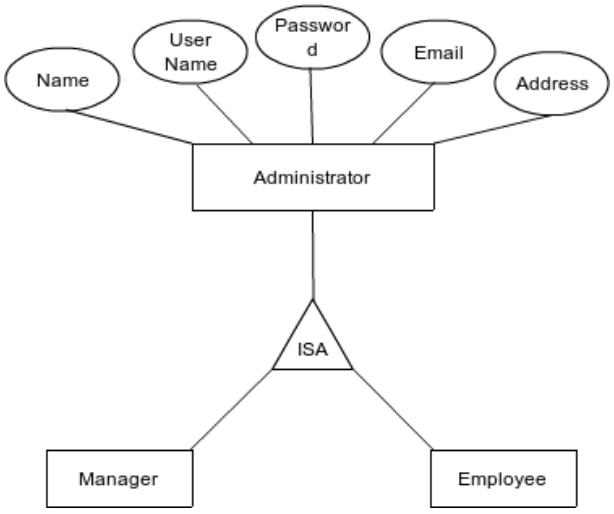


Figure 1: E-R Model

.

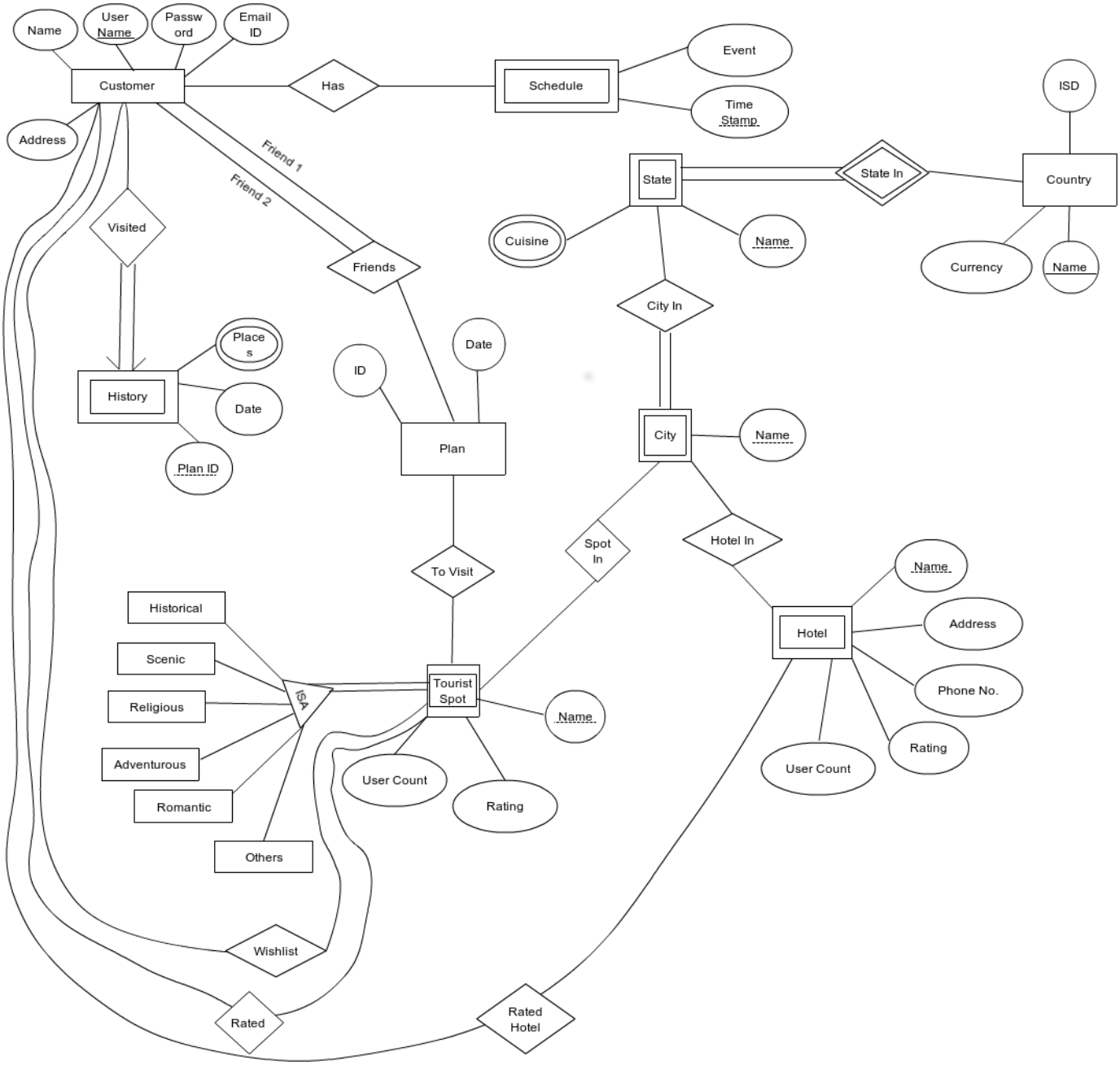


Figure 2: E-R Model

## 9 Test Plan

- Search for a tourist place by country, state or city :
  In this case, no edge cases need to be taken care of as the user is not allowed to query manually.

- Search for a tourist place by its name :
  If the user enters a wrong name (one that does not exist in the database), we will redirect her to an error page showing the name entered is invalid. This page will have an option to go back to the search page.

- Search for hotels by city or tourist spot :
  If the user enters a wrong hotel name (one that does not exist in the database corresponding to the place or city), we will redirect her to an error page showing the name entered is invalid. This page will have an option to go back to the search page.

- Search for speciality cuisines of the region :
  In this case, we will show her the cuisine results only for that particular region (no option to manually search will be provided to the user). Hence, no exceptions to handle.

- Rate places visited and hotels stayed in :
  If a user decides not to rate a hotel or place, there should not be an error. The rating will be on a scale of 10. Also, same user will not be allowed to rate a place or hotel multiple times.

- Maintain and modify a wishlist for the places (tourist spot) she wants to visit :
  On delete of a particular entry from wishlist, the remaining entries in the wishlist should remain intact.

- Prepare and modify a travel plan for her tour :
  In case of a group travel plan (Friends plan), if a particular customer modifies her plan, the plan must be automatically updated in the other friends' plans too. Plan must be consistent for all users. If two users attempt to modify group travel plan, it must not be allowed. We will use synchronization for this.

- Add a user to a group plan :
  If a customer is added to a group plan without her permission, she must not be added to the plan. Repeated addition of the same customers will throw an error message.

- Maintain a history which records all the details of her trip :
  Since history is never manually modified, no boundary cases will come up.

- Update personal information :
  While updating email address, if the customer enters an already existing email id, she will be prompted to enter the email address again. Also, if she leaves blank the cells corresponding to name, password or email id, we will ask her to enter valid entries.

- Modify the tourism database (for administrator and employees) :
  If the admin deletes a city from a database, the corresponding tourist spots should also be deleted. All dependencies must be dealt with properly.
  If a new city is added by an employee and it already exists in the database, we will prompt her with an error.

- Create logins for recruited employees (for administrator) :
  Similar to point Update personal information.
- Updating/creating/maintaining plan :
  Date and time of a new plan, if given before the current time, the user should be prompted with an error and asked to enter a valid time/date.

# 10 Schema Design, Screen Design, Functional Dependencies and Assertions

## Schema Design

### 10.0.1 Entities

- Country(Name PRIMARY KEY, Currency, ISD)
- State(Country(Reference to Country.Name), Name(Discriminator), Cuisine, STDCode)
- City(Country(Reference to Country.Name), State(Reference to State.Name), Name(Discriminator), Pincode UNIQUE)
- Customer(Username PRIMARY KEY, Name NOT NULL, Password NOT NULL, EmailID NOT NULL, Address)
- Schedule(CustomerID (Reference to Customer.Username),Event NOT NULL, TimeStamp(Discriminator) NOT NULL)
- History(CustomerID (Reference to Customer.Username), PlanID (DISCRIMINATOR), Date NOT NULL, PlaceID NOT NULL)
- HistoryPlaces(PlaceID(Reference to History.PlaceID), CountryName(Reference to Country.Name), StateName(Reference to State.Name), CityName(Reference to City.Name), SpotName(Reference to TouristSpotName))
- Plan(ID PRIMARY KEY)
- Hotel(CountryName(Reference to Country.Name), StateName(Reference to State.Name), CityName(Reference to City.Name), HotelName(Discriminator), Address NOT NULL, PhoneNo., NumberOfHits, Rating)

### 10.0.2 IS-A Relationships

- For Administration
  - E/R Approach
    - Administrator(Name NOT NULL, Username PRIMARY KEY, Password NOT NULL, Role NOT NULL, EmailID , Address)
    - Manager(ID (Reference to Administrator.Username))
    - Employee(ID (Reference to Administrator.Username))
  - OO Approach
    - Administrator(Name NOT NULL, Username PRIMARY KEY, Password NOT NULL, Role NOT NULL, EmailID , Address)

    * Manager(Name NOT NULL, Username PRIMARY KEY, Password NOT NULL, Role NOT NULL, EmailID , Address)
    * Employee(Name NOT NULL, Username PRIMARY KEY, Password NOT NULL, Role NOT NULL, EmailID , Address)

- For Tourist-Spot
  - E/R Approach
    * TouristSpot(CountryName(Reference to Country.Name), StateName(Reference to State.Name), CityName(Reference to City.Name), Name(Discriminator), Rating, Description)
    * Historical(CountryName(Reference to Country.Name), StateName(Reference to State.Name), CityName(Reference to City.Name), SpotName(Discriminator), YearReference)
    * Scenic(CountryName(Reference to Country.Name), StateName(Reference to State.Name), CityName(Reference to City.Name), SpotName(Discriminator))
    * Historical(CountryName(Reference to Country.Name), StateName(Reference to State.Name), CityName(Reference to City.Name), SpotName(Discriminator), Religion)
    * Adventurous(CountryName(Reference to Country.Name), StateName(Reference to State.Name), CityName(Reference to City.Name), SpotName(Discriminator), AdventureType)
    * Others(CountryName(Reference to Country.Name), StateName(Reference to State.Name), CityName(Reference to City.Name), SpotName(Discriminator), Speciality NOT NULL)
  - OO Approach
    As we have 5 subclasses for TouristSpot Class, here for sake of convenience and readability we are describing only some of the relations, like
    * Historical(CountryName(Reference to Country.Name), StateName(Reference to State.Name), CityName(Reference to City.Name), SpotName(Discriminator), YearReference, Rating, Description)
    * HistReli(CountryName(Reference to Country.Name), StateName(Reference to State.Name), CityName(Reference to City.Name), SpotName(Discriminator), YearReference, Religion, Rating, Description)
    * AdvenHistScenic(CountryName(Reference to Country.Name), StateName(Reference to State.Name), CityName(Reference to City.Name), SpotName(Discriminator), AdvcentureType, YearReference, Rating, Description)
    * And many more

#### 10.0.3 Many-to-many Relationships

Here in our Database there are only many-to-many relationships that are "Friends" and "ToVisit", "RatedHotel", "RatedSpot" and "Wishlist". Here Plan Entity is modified as per stated in the problem statement corresponding to many-to-many relationships Friends and ToVisit . Above Plan Entity should be ignored. Also "Customer" Relation is also modified corresponding to "Wishlist","RatedSpot" and "RatedHotel". Above "Plan" and Customer Entities should be ignored.

- Plan(CustomerID(Reference to Customer.Username), PlanID, GroupID NOT NULL, Date NOT NULL, CountryName(Reference to Country.Name), StateName(Reference to State.Name), CityName(Reference to City.Name), SpotName(Reference to TouristSpotName)). NumberofGroup corresponds to multi-attribute field

- Group(GroupID(Reference to Friends.GroupID), Friend(Reference to Customer.ID))
- Customer(Username PRIMARY KEY, Name NOT NULL, Password NOT NULL, EmailID NOT NULL, Address, HotelRating, SpotRating, HotelName(Referenced to Hotel.Name), SpotName(Referenced to TouristSpot.Name), CityName(Referenced to City.Name), StateName(Referenced to State.Name), CountryName(Referenced to Country.Name))

  **Assertions**

  - STD codes for different states are different in a country.
  - Pincodes are different for different cities in a country.
  - In the Schedule/Plan relation a user can update/enter event corresponds to only after the current timestamp.
  - Rating of a hotel or a tourist spot should be consistent with the rating data on updation of its rating by the user. i.e. the current rating should be updated if a user update or rate a spot or hotel.

## Functional Dependecies

- For Customer Relation
  - Username → password, name, EmailID, Address.
  - EmailID → Username
- For History Relation
  - PlanID, Username → PlaceID, Date
- For Schedule Relation
  - Username, TimeStamp → Event
- For Country Relation
  - Name → Currency, ISD
- For State Relation
  - CountryName, StateName → Cuisine
- For City Relation
  - CountryName, StateName, CityName → CountryName, StateName, CityName
- For Hotel Relation
  - CountryName, StateName, CityName,HotelName → Address,PhoneNo., Rating, NumberOfHits
- For RatedHotel Relation
  - CountryName, StateName, CityName,HotelName,Username → Rating

- For TouristSpot Relation
  - CountryName, StateName, CityName, Name → Rating, NumberOfHits,
  - Given Cusatomer.Username → WishlistID
  - CustomerID, CountryName, StateName, CityName, TouristSpot.Name → User.SpotRating, Description
  - CustomerID, CountryName, StateName, CityName, Hotel.Name → User.HotelRating
- For Plan Relation
  - CustomerID, PlanID → GroupID
  - PlanID, Date → SpotID

### Screen Design

Screen Design are to be found in "screen$_{d}esign - 2013 - 10 - 13.zip$"$inthefolder$.