Functional Specification, E-R Model, Test Plan, Schema Design and Screen Design for Database and Information Systems Project

October 20, 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

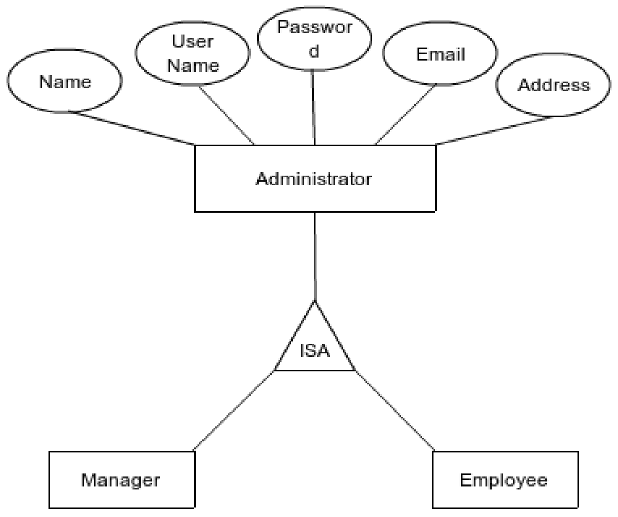


Figure 1: E-R Model

.

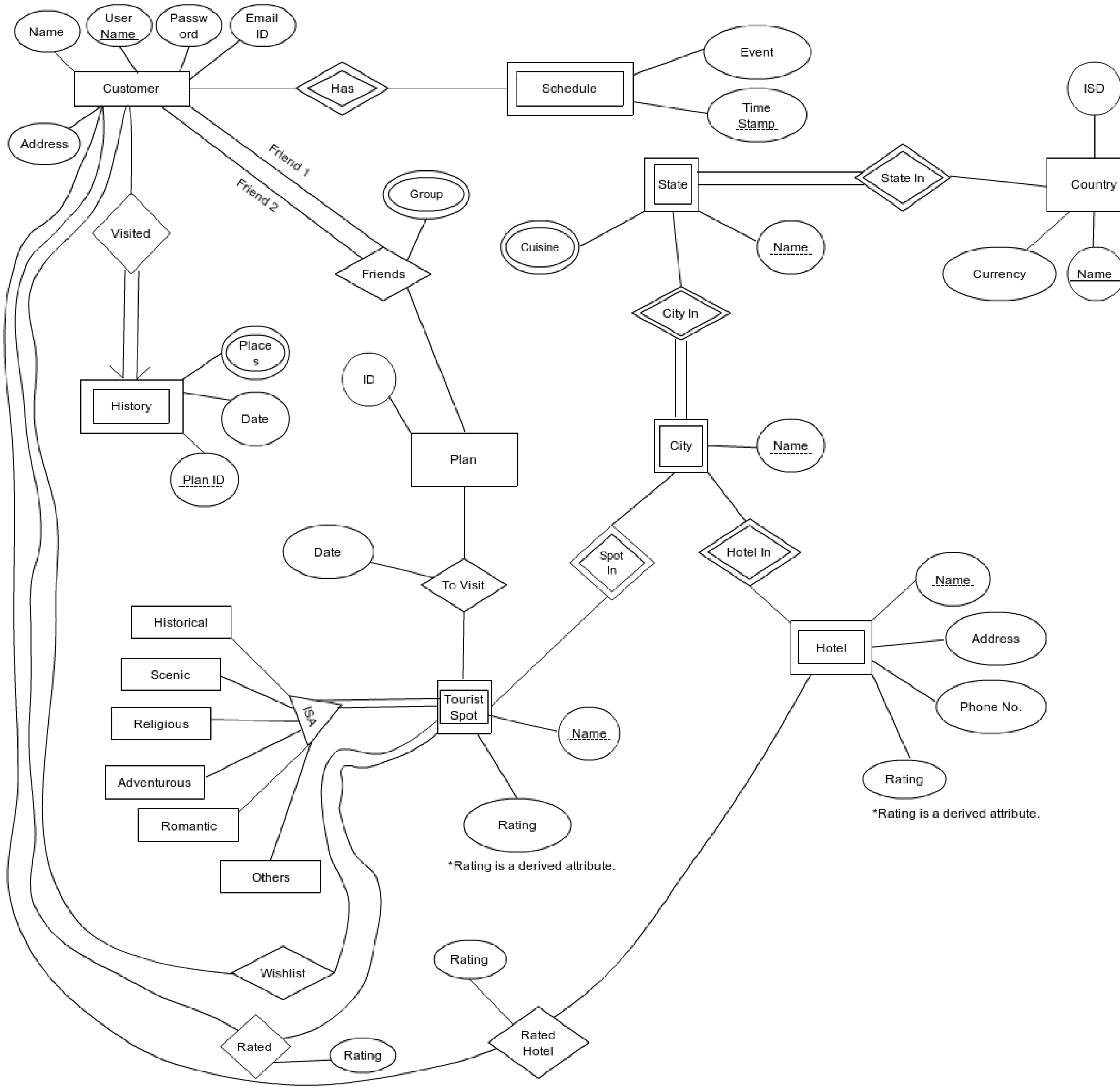


Figure 2: E-R Model

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

10.1 Schema Design

10.1.1 Entities

- Country(Name PRIMARY KEY, Currency, ISD)
- State(Country(Reference to Country.Name), Name(Discriminator), Cuisine, STDCode)
- City(Country(Reference to Country.Name), State , 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 , CityName , SpotName(Reference to TouristSpotName))
- Plan(ID PRIMARY KEY)
- Hotel(CountryName(Reference to Country.Name), StateName , CityName , HotelName(Discriminator), Address NOT NULL, PhoneNo., NumberOfHits, Rating)

10.1.2 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 , CityName , Name(Discriminator
      Rating, Description)
    - Historical(CountryName(Reference to Country.Name), StateName , CityName , SpotName(Discrimina
      YearReference)
    - Scenic(CountryName(Reference to Country.Name), StateName , CityName , SpotName(Discriminator)
    - Historical(CountryName(Reference to Country.Name), StateName , CityName , SpotName(Discrimina
      Religion)
    - Adventurous(CountryName(Reference to Country.Name), StateName , CityName , SpotName(Discrim
      AdventureType)
    - Others(CountryName(Reference to Country.Name), StateName , CityName , 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 , CityName , SpotName(Discrimina
      YearReference, Rating, Description)
    - HistReli(CountryName(Reference to Country.Name), StateName , CityName , SpotName(Discriminato
      YearReference, Religion, Rating, Description)
    - AdvenHistScenic(CountryName(Reference to Country.Name), StateName , CityName , SpotName(Discriminator), AdvcentureType, YearReference, Rating, Description)
    - And many more

10.1.3 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 , CityName , 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, EmailID NOT NULL, RSCityName(Referenced to City.Name), RSStateName(Referenced to State.Name), RSCountryName(Referenced to Country.Name), SpotName(Referenced to TouristSpot.Name), SRating, RHCityName(Referenced to City.Name), RHStateName(Referenced to State.Name), RHCountryName to Country.Name), HotelName(Referenced to Hotel.Name), HRating, WCityName(Referenced to City.Name), WStateName(Referenced to State.Name), WCountryName(Referenced to Country.Name), WSpotName(Referenced to TouristSpot.Name))

10.2 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.

10.3 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

10.4 Normalization

- Customer(Username PRIMARY KEY, Name NOT NULL, Password NOT NULL, EmailID NOT NULL, Address, EmailID NOT NULL, RSCityName(Referenced to City.Name), RSStateName(Referenced to State.Name), RSCountryName(Referenced to Country.Name), SpotName(Referenced to TouristSpot.Name), SRating, RHCityName(Referenced to City.Name), RHStateName(Referenced to State.Name), RHCountryName to Country.Name), HotelName(Referenced to Hotel.Name), HRating, WCityName(Referenced to City.Name), WStateName(Referenced to State.Name), WCountryName(Referenced to Country.Name), WSpotName(Referenced to TouristSpot.Name)).

  We have the following FDs from the above relation :

  - Username → Name, Password, Address, EmailID
  - Username,RSCityName, RSStateName, RSCountryName, SpotName → SpotRating
  - Username, RHCityName, RHStateName, RHCountryName, HotelName → HotelRating
  - Username, WCityName, WStateName, WCountryName → WSpotName

  Now we can see that not a single FD is trivial and none of the LHS of the FD is super-key, this implies that the above relation is not in BCNF. Following the rules for BCNF decomposition, we get the following to relations :

  - R1(Username, Password, Name, Address, EmailID)
  - R2(RSCityName, RSStateName, RSCountryName, SpotName, SRating, RHCityName, RHStateName, RHCountryName, HotelName, HRating, WCityName, WStateName, WCountryName, WSpotName).

  Now R2 is not in BCNF while R1 is. FD no.1 is still preserved. Applying BCNF decomposition to R2 based on FDs no. 2,3,4, altogether we get four relations, which are:

  - R1(Username, Password, Name, Address, EmailID)
  - R2(Username, RSCityName, RSStateName, RSCountryName, SpotName, SRating)
  - R3(Username, RHCityName, RHStateName, RHCountryName, HotelName, HRating)
  - R4(Username, WCityName, WStateName, WCountryName, WSpotName)

  Since all the FDs remain preserved after this decomposition, we need not revert to 3NF as there are no added redundancies. Also there is no need for 4NF decomposition.

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

  We have the following FDs from the above relation :

  - CustomerID, PlanID → GroupID

Now, since the FD is not trivial and nor is (CustomerID, PlanID) a super-key for the relation, we can decompose it to BCNF. Following the rules for BCNF decomposition, we get the following to relations :

- R1(CustomerID, PlanID, GroupID). This relation is consistent with the friends relation in our E/R model and is in BCNF.
- R2(CustomerID, PlanID, CountryName, StateName, CityName, SpotName, Date).

In this relation there are no non-trivial FDs possible as the super-key consists of all the attributes of the relation taken together. Thus, we have one trivial FD. Hence, this relation too is in BCNF. R2 corresponds with ToVisit relation in the E/R model.
Since all the FDs remain preserved after this decomposition, we need not revert to 3NF as there are no added redundancies.
We have the following multivalued dependencies from R2 :

- CustomerID, PlanID, Date $\rightarrow$ CountryName, StateName, CityName, SpotName

This FD implies that for a given (CustomerID, PlanID, Date) tuple, a customer can have many (CountryName, StateName, CityName, SpotName) tuples. He can visit many places on a given date in a given plan. Now, we apply the 4NF decomposition algorithm to this FD. We get the following relations:

- R3(CustomerID, PlanID, Date , CountryName, StateName, CityName, SpotName). R3 is obviously in 4NF.
- R4(CustomerID, PlanID, Date, GroupID). For R2, we have only trivial FDs. Hence, R4 too is in 4NF.

However, breaking into R3 and R4 reduces no redundancy, infact, redundancy is increased and hence, this 4NF decomposition is trivial. We conclude that R2 is in 4NF too.

Further Modifications

The E/R diagram shows an IS-A relation for the TouristSpot entity. We have modified this. Since a tourist spot can be under more than one IS-A categories, a lot of redundancies will creep in once we build the database. Hence, for the sake of efficient design, we have introduces a Type multivalued attribute for the Tourist Spot entity. This will contain names of all the former IS-A categories where that particular spot falls like Historical or Religious or Historical and Religious.

10.5 Final Schema Design

Following are the schemas which we are going to use in the project

- Country(Name, Currency, ISD)
- State(Country FOREIGN KEY Country.Name), Name, Cuisine, STDCode)
- City(Country(FOREIGN KEY Country.Name), State , Name, Pincode)
- TouristSpot(CountryName(FOREIGN KEY Country.Name), StateName, CityName, SpotName, Rating, Description, TypeID NOT NULL)

- TouristSpotType(TypeID NOT NULL, Type(1 or more than 1 combination of ’Historical’, ’Adventerous’, ’Religious’ and ’Scenic’ or ’Other’))
- Customer(Username, Password, Name, Address, EmailID)
- Administrator(Username, Password, Name, Address, EmailID, Role(’Manager’ or ’Employee’))
- Schedule(Username(FOREIGN KEY Customer.Username), TimeStamp, Event NOT NULL)
- History(CustomerID(FOREIGN KEY Customer.Username), PlanID, Date, PlaceID NOT NULL)
- HistoryPlaces(PlaceID(FOREIGN KEY History.PlaceID), CountryName(FOREIGN KEY Country.Name), StateName , CityName , SpotName)
- Plan(ID)
- SpotRating(Username(FOREIGN KEY Customer.Username), RSCityName , RSStateName , RSCountryName
SRating)
- HotelRating(Username(FOREIGN KEY Customer.Username), RHCityName , RHStateName , RHCountryNa
HRating)
- WishList(Username(FOREIGN KEY Customer.Username), WCityName , WStateName , WCountryName(FOR
KEY Country.Name), WSpotName)
- ToVisit(CustomerID(FOREIGN KEY Customer.username), PlanID(FOREIGN KEY Plan.ID), CountryName(F
KEY Country.Name), StateName, CityName , SpotName, Date).
- Friends(CustomerID(FOREIGN KEY Customer.username), PlanID(FOREIGN KEY Plan.ID), GroupID)
- Group(GroupID(FOREIGN KEY Friends.GroupID), Friend(FOREIGN Key Customer.ID))
- Hotel(CountryName(FOREIGN KEY Country.Name), StateName, CityName, HotelName, Address NOT NULL, PhoneNo NOT NULL, NumberOfHits NOT NULL, Rating NOT NULL)

10.6 Screen Design

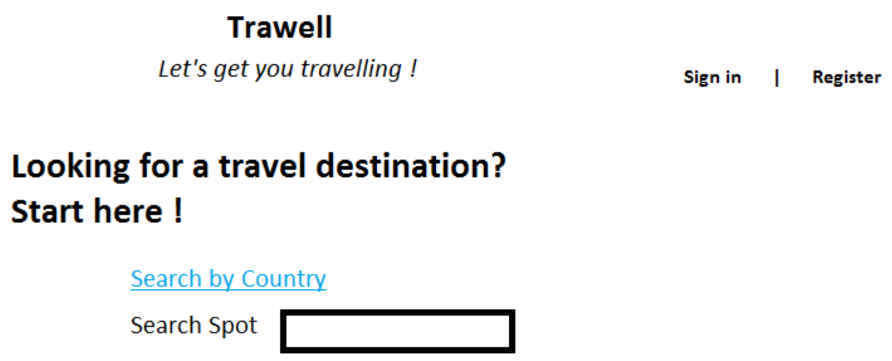


Figure 3: Home Screen

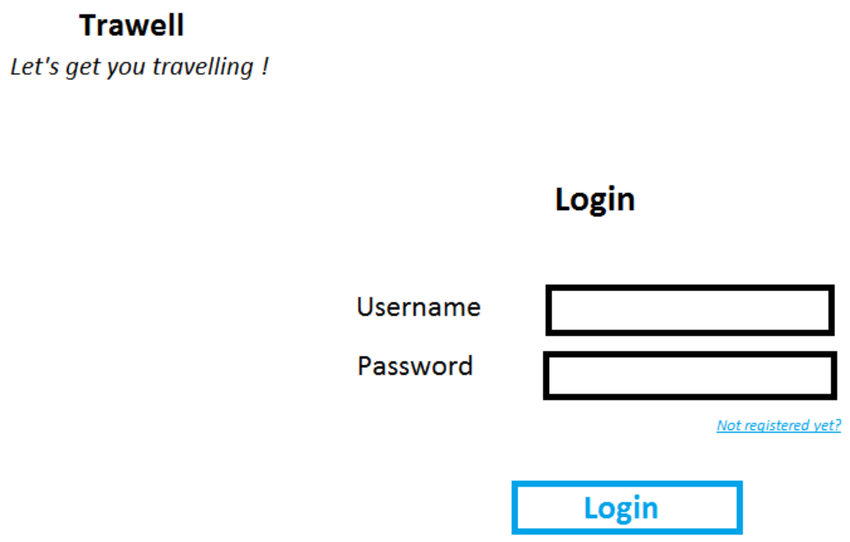


Figure 4: Login page

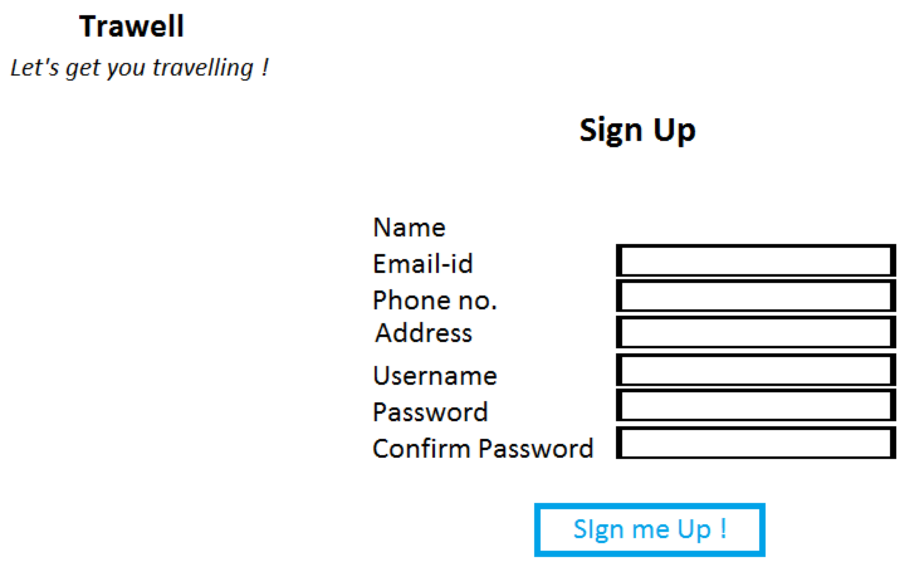


Figure 5: Registration page

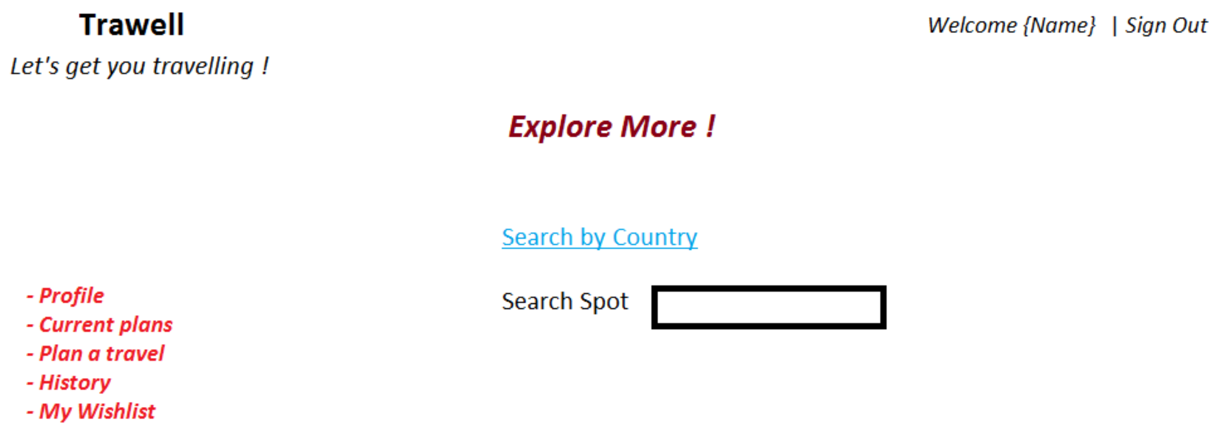
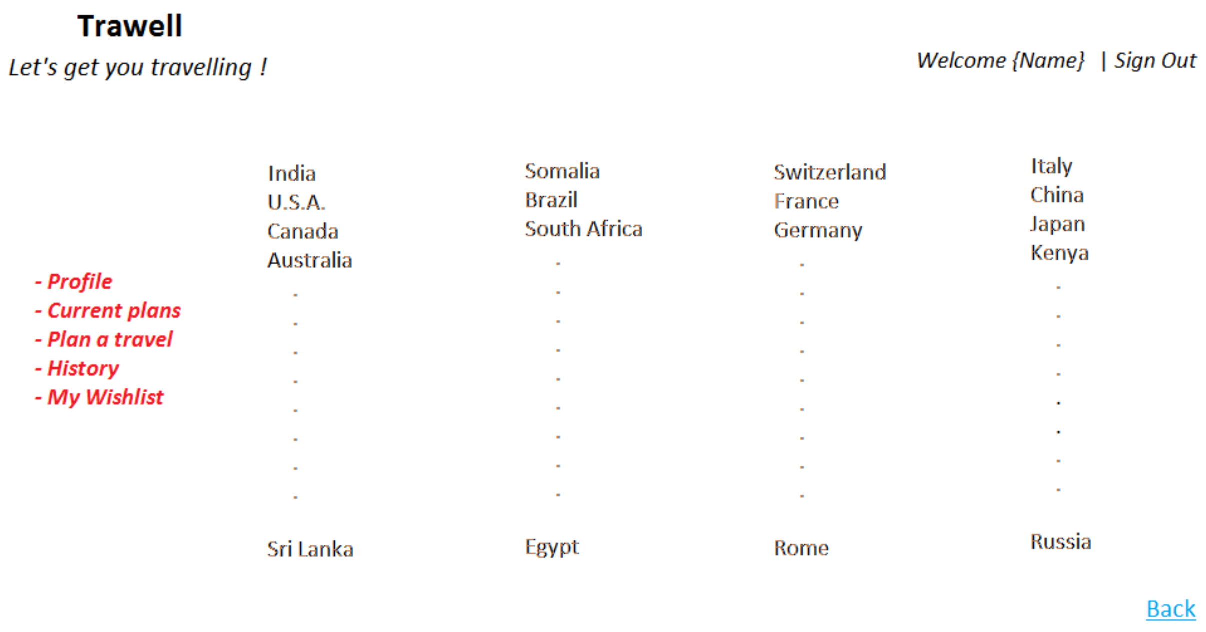


Figure 6: Logged In page



Trawell

Let's get you travelling !

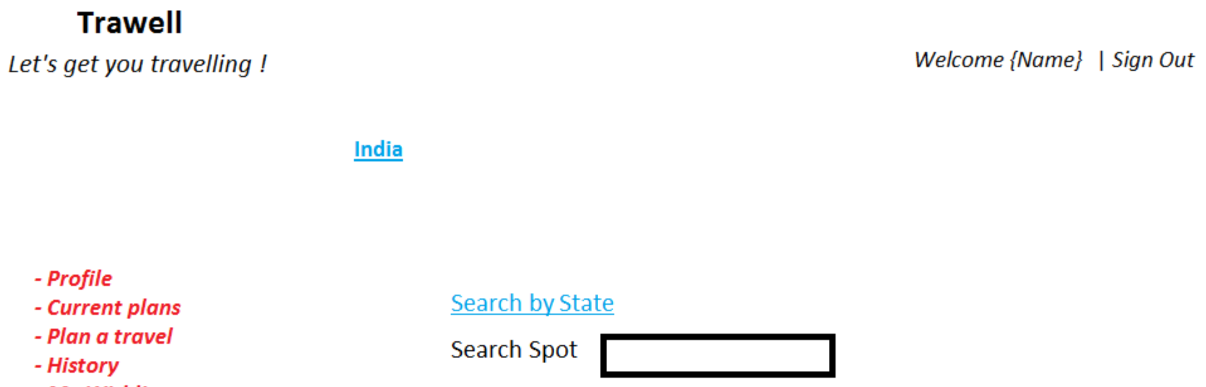
Welcome {Name} | Sign Out

- Profile
- Current plans
- Plan a travel
- History
- My Wishlist

India	Somalia	Switzerland	Italy
U.S.A.	Brazil	France	China
Canada	South Africa	Germany	Japan
Australia	.	.	Kenya
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
Sri Lanka	Egypt	Rome	Russia

Back

Figure 7: Search by country → Results



Trawell

Let's get you travelling !

Welcome {Name} | Sign Out

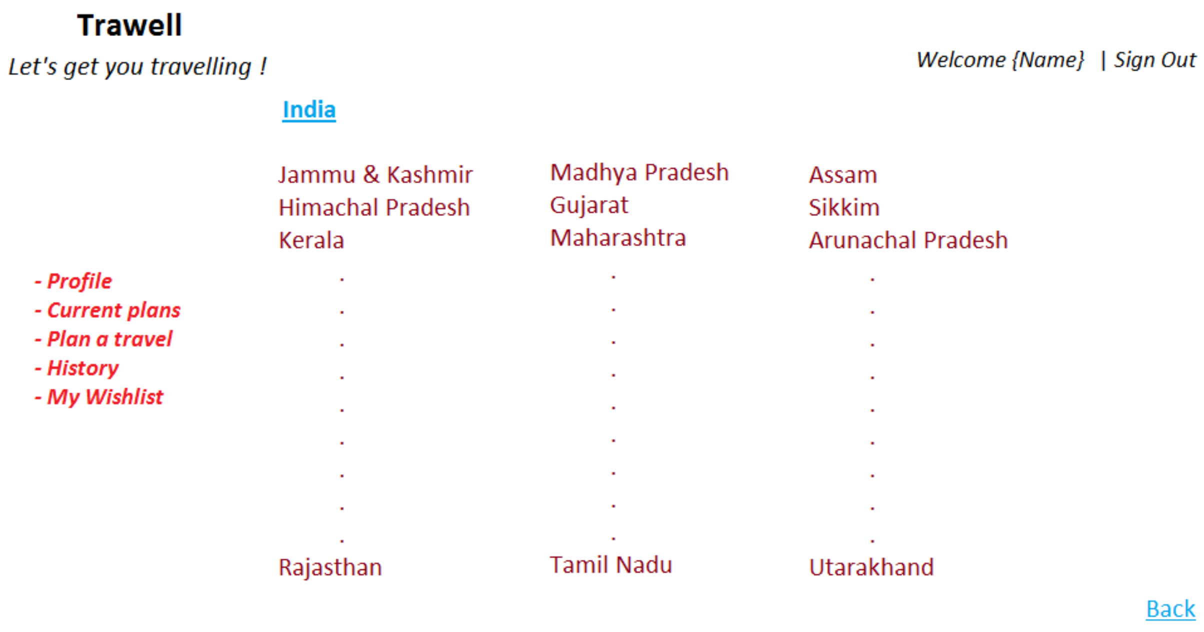
India

- Profile
- Current plans
- Plan a travel
- History
- My Wishlist

Search by State

Search Spot

Figure 8: Search by state of the country selected



Trawell

Let's get you travelling !

Welcome {Name} | *Sign Out*

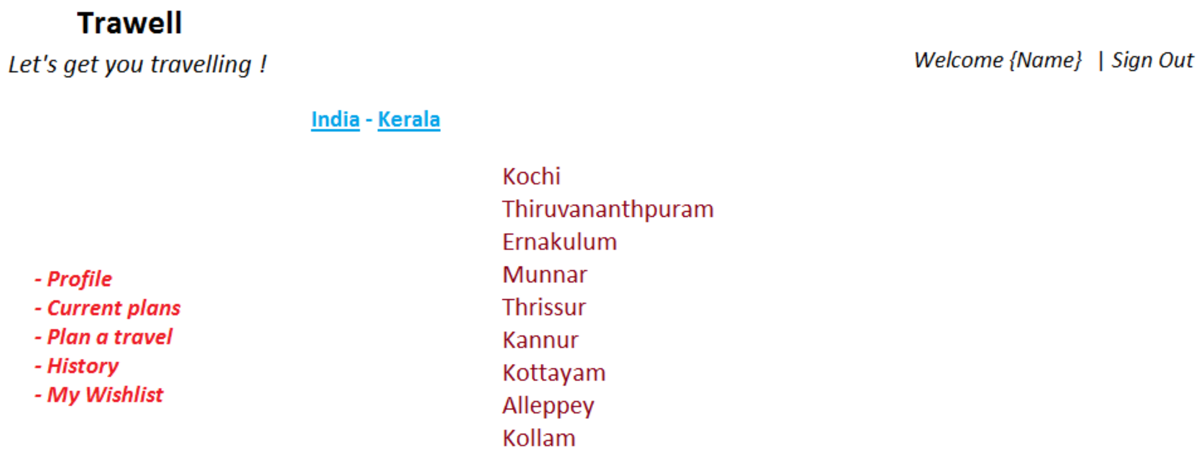
India

Jammu & Kashmir	Madhya Pradesh	Assam
Himachal Pradesh	Gujarat	Sikkim
Kerala	Maharashtra	Arunachal Pradesh
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
Rajasthan	Tamil Nadu	Utarakhand

- Profile
- Current plans
- Plan a travel
- History
- My Wishlist

Back

Figure 9: States of selected country



Trawell

Let's get you travelling !

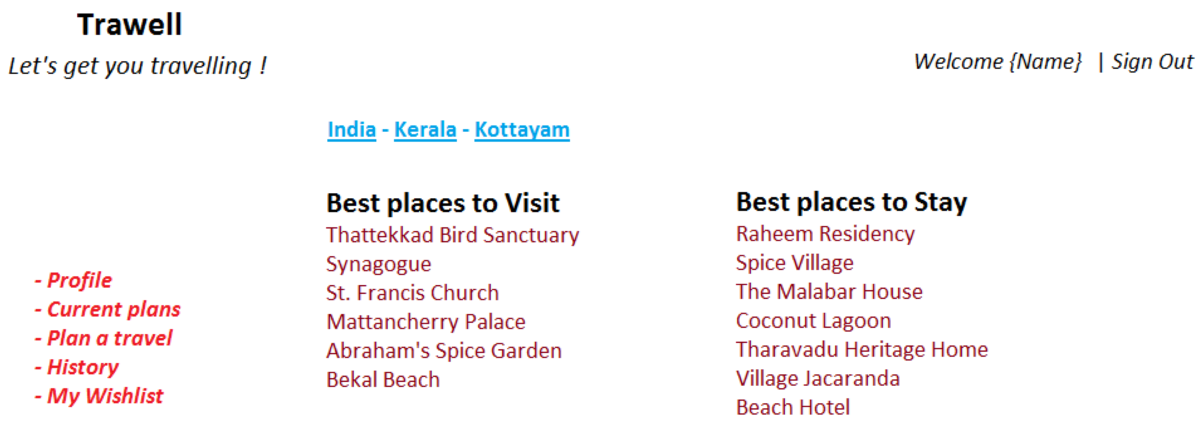
Welcome {Name} | *Sign Out*

India - Kerala

Kochi
Thiruvananthpuram
Ernakulum
Munnar
Thrissur
Kannur
Kottayam
Alleppey
Kollam

- Profile
- Current plans
- Plan a travel
- History
- My Wishlist

Figure 10: Popular cities of the state selected


Trawell

Let's get you travelling !

Welcome {Name} | Sign Out

India - Kerala - Kottayam

- *Profile*
- *Current plans*
- *Plan a travel*
- *History*
- *My Wishlist*

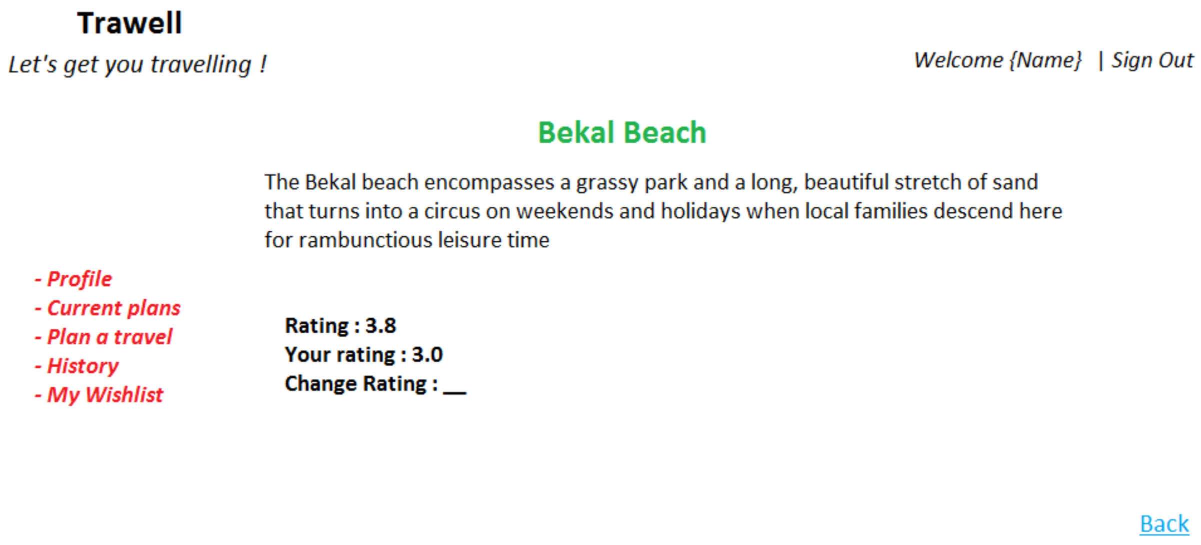
Best places to Visit

Thattekkad Bird Sanctuary
Synagogue
St. Francis Church
Mattancherry Palace
Abraham's Spice Garden
Bekal Beach

Best places to Stay

Raheem Residency
Spice Village
The Malabar House
Coconut Lagoon
Tharavadu Heritage Home
Village Jacaranda
Beach Hotel

Figure 11: Popular Tourist Spots & Hotels of the selected city


Trawell

Let's get you travelling !

Welcome {Name} | Sign Out

Bekal Beach

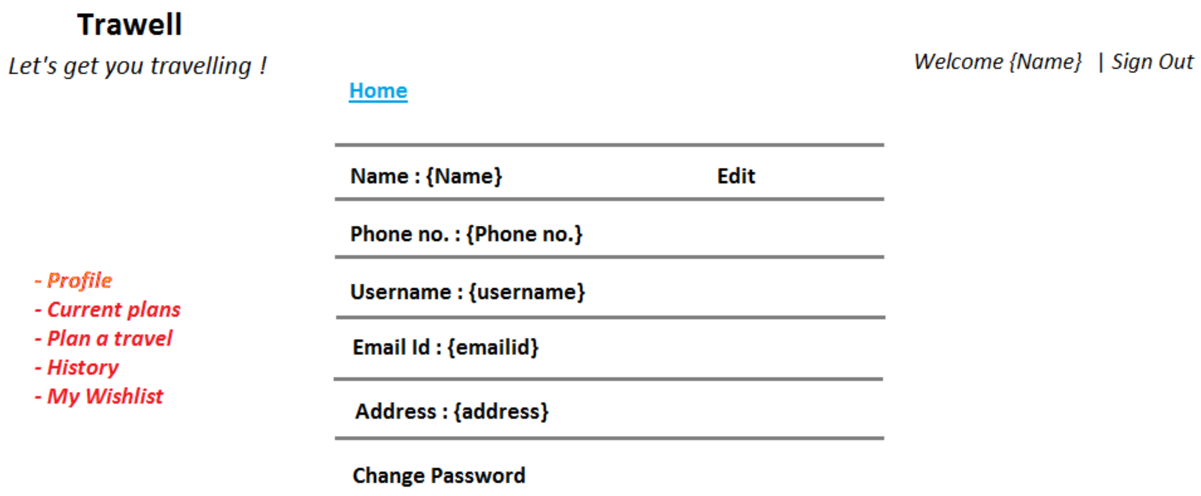
The Bekal beach encompasses a grassy park and a long, beautiful stretch of sand that turns into a circus on weekends and holidays when local families descend here for rambunctious leisure time

- *Profile*
- *Current plans*
- *Plan a travel*
- *History*
- *My Wishlist*

Rating : 3.8
Your rating : 3.0
Change Rating : __

Back

Figure 12: Description of the selected Spot or Hotel



Trawell

Let's get you travelling !

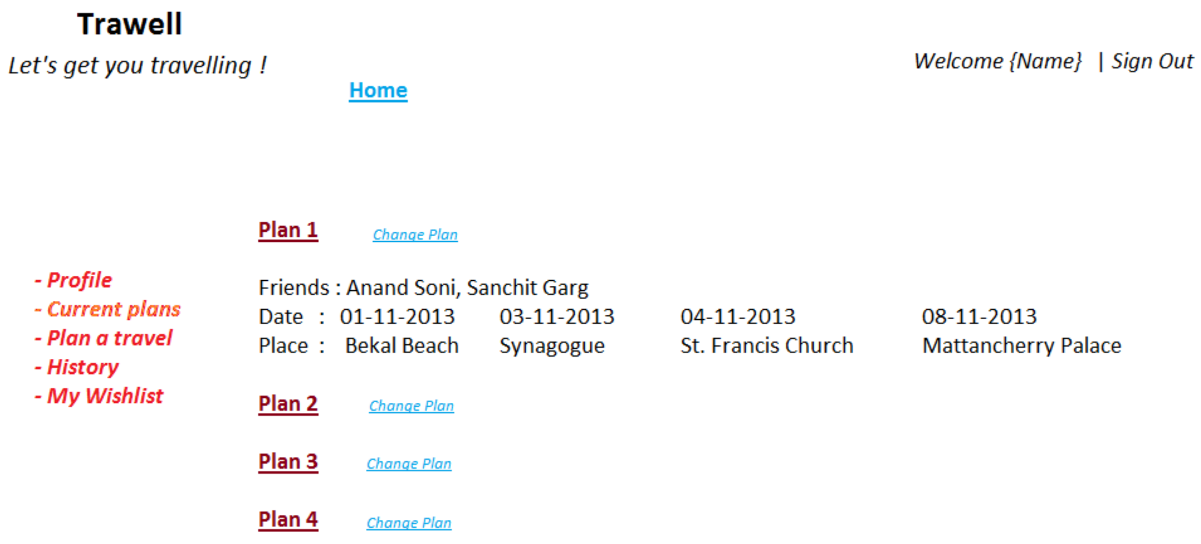
Welcome {Name} | *Sign Out*

Home

- Profile
- Current plans
- Plan a travel
- History
- My Wishlist

Name : {Name}	Edit
Phone no. : {Phone no.}	
Username : {username}	
Email Id : {emailid}	
Address : {address}	
Change Password	

Figure 13: User Profile



Trawell

Let's get you travelling !

Welcome {Name} | *Sign Out*

Home

- Profile
- Current plans
- Plan a travel
- History
- My Wishlist

Plan 1 Change Plan

Friends : Anand Soni, Sanchit Garg

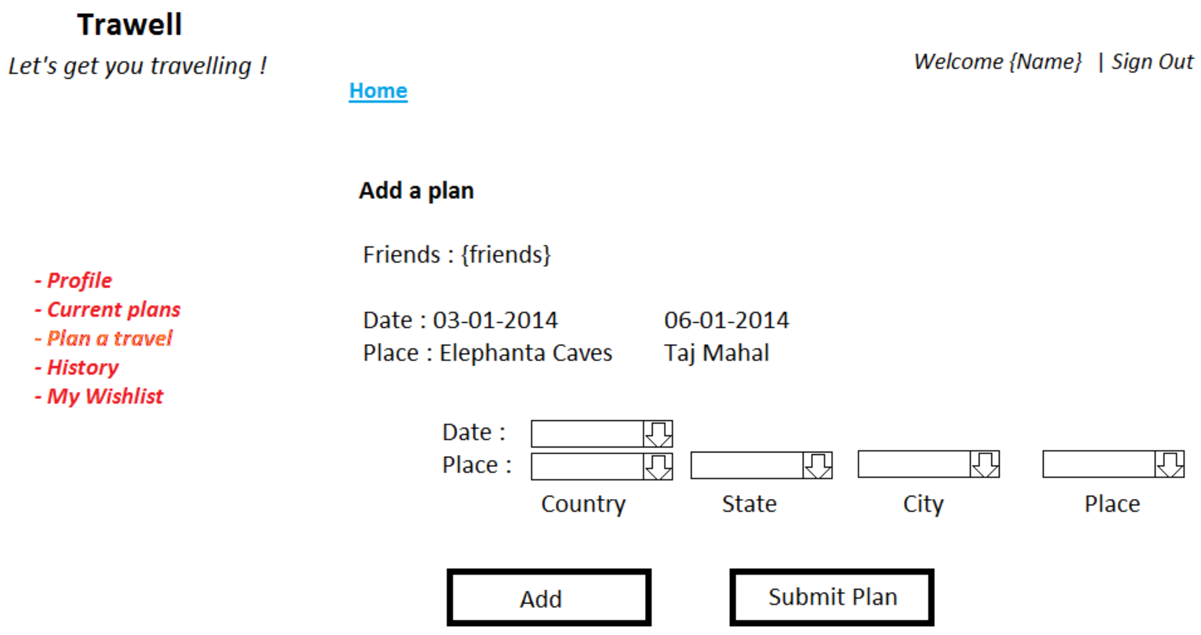
Date :	01-11-2013	03-11-2013	04-11-2013	08-11-2013
Place :	Bekal Beach	Synagogue	St. Francis Church	Mattancherry Palace

Plan 2 Change Plan

Plan 3 Change Plan

Plan 4 Change Plan

Figure 14: Current plans of the User


Trawell

Let's get you travelling !

Welcome {Name} | Sign Out

Home

- Profile
- Current plans
- Plan a travel
- History
- My Wishlist

Add a plan

Friends : {friends}

Date :	03-01-2014	06-01-2014
Place :	Elephanta Caves	Taj Mahal

Date :

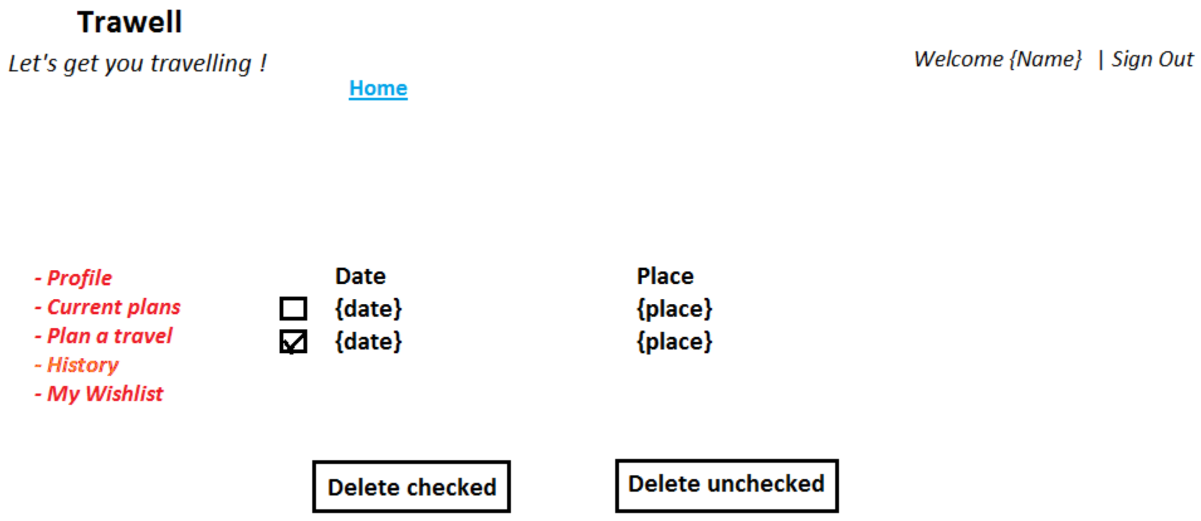
Place :

Country State City Place

Add

Submit Plan

Figure 15: Add a new Plan


Trawell

Let's get you travelling !

Welcome {Name} | Sign Out

Home

- Profile
- Current plans
- Plan a travel
- History
- My Wishlist

	Date	Place
☐	{date}	{place}
☑	{date}	{place}

Delete checked

Delete unchecked

Figure 16: History of the travel of the User

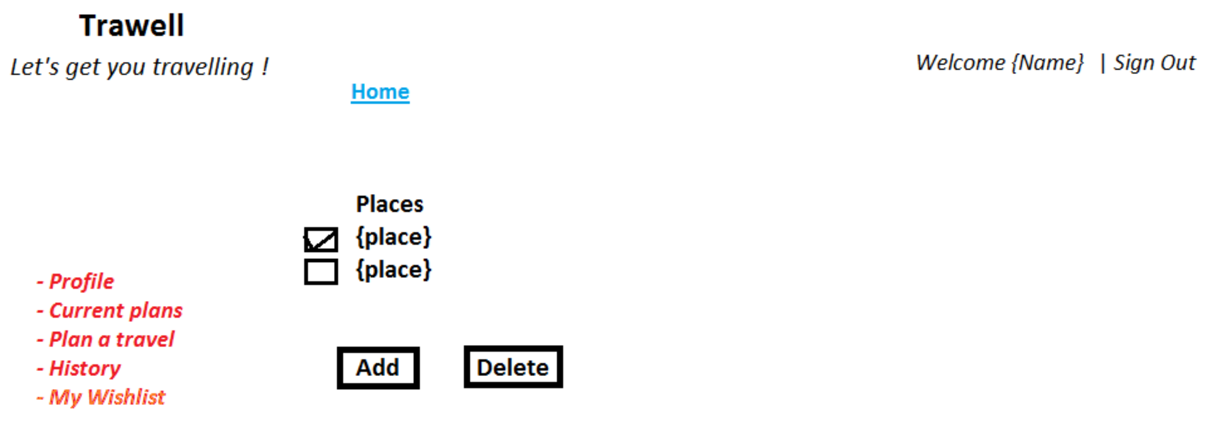


Figure 17: User's Wishlist