

# CS 387 Database and Information Systems Lab

## Project Proposal - ConsumerConnect

Astha Agarwal (110050018)  
Anmol Garg (110050020)  
Deepali Adlakha (11D170020)

October 13, 2013

## 1 Introduction - Application Domain

As our CS 387 project, we have planned to make an online service portal, whereby people can interact with service providers and other customers to seek opinion and decide the appropriate service provider/service they want to use. Here, 'customers' refer to other users on the database, who also interact in a similar manner.

On the other side, service providers will also be able to interact with users to know their potential customers. They can give details about the services they provide, and answer customer queries.

The project involves crowdsourcing, search and social networking.

## 2 Functional Specification

Broadly, there are two types of users of our application - customers and service providers. We plan to provide a portal for them to interact. At present, we have not added any administrators, though this is a feature that can be added if required. Here is a brief description of the facilities which will be provided to the users:

### 2.1 Customer

1. **Personal Details** - Users can provide their personal information like name, address, age, gender etc.
2. **Reviews** - Users can provide reviews and rate the service providers according to their experience. (Validating the experience can be seen as a further extension of the project.) Also, they can upvote or downvote others' reviews. Users will earn points when their reviews are upvoted. Their rank improves as the number of points increases.
3. **Friends** - Users can see the details of other users tagged as friends and reviews posted by them.
4. **Messaging** - Users can send a message to other customers to seek opinion or otherwise.
5. **Past Experiences** - User can refer to his experience in the past, and make an informed decision accordingly.
6. **Wishlist** - User can maintain a wishlist of the services required by him for future reference.

7. **Appointments** - Customers can book an appointment with a service provider for a particular service.
8. **Bidding** - The wishlist of customers can be viewed by a service provider, and he can post a bid for the corresponding service. Customer can then make a decision by comparing various prices and options depending on the response of service providers.

## 2.2 Service Provider

1. **Provider Details** - Users can provide their personal information like name, address, age, gender etc.
2. **Service Details** - User can add the details of service provided by him such as service type, rates, availability etc.
3. **Rating** - An average rating based on reviews given by customers to his services.
4. **Reviews** - The reviews given by his customers will be visible to other willing customers for reference.
5. **Bidding and Appointments** - As explained above, these are interactions between service providers and customers, which help them communicate well with each other.

## 3 Data Model Specification

In our ER Model Diagram, in addition to the popular convention, we have used separate colour for the weak relationship sets

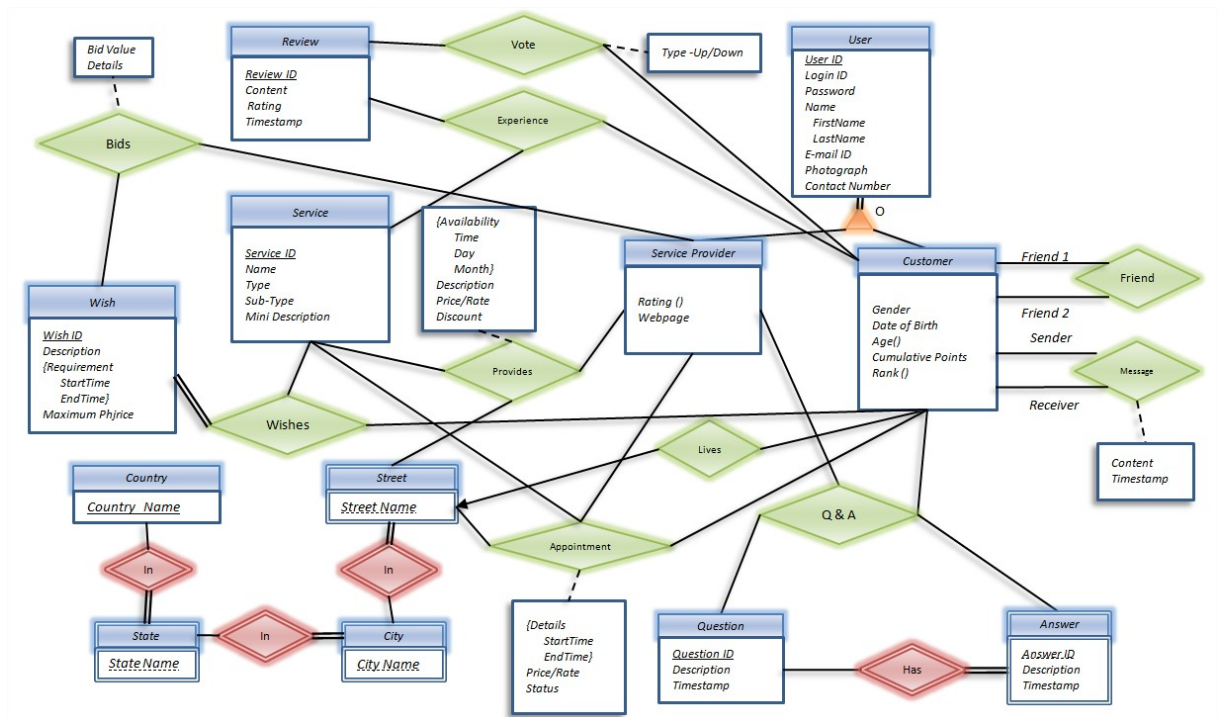
### 3.1 Entities

1. *User* of our application - The user has been allotted a User ID Login ID, Password, Name, E-mail ID, Photograph and Contact Number. Two entities are inherited from the *User* - *Customer* and *Service Provider*.
2. *Customer*, is one possible profile for a *User* and is allotted a customerID (Primary Key). The customer has a Gender, Date of Birth, Age, (derived attribute), Cumulative Points (earned through the votes on his reviews to Services) and Rank (derived from his Cumulative Points).
3. *Service Provider*, is the second possible profile for the *User* and is allotted a Service Provider ID (Primary Key). *Service Provider* has a Rating, a quantitative measure of the quality of services he provides. (derived attribute). He can also mention his Webpage.
4. *Service* - It has a Service ID(Primary Key) and Name. The Type of Service is a broad categorisation (eg. Doctor, Repair) and Sub-Type is a specific category(eg. Cardiology, Cycle Repair). There can be a Mini Description associated to the service.
5. *Review*, given by a *Customer* to a *Service* provided by a *Service provider*. It has a primary key Review ID. There is a Content of the Review, a Rating which the customer gives to the service (and service provider), and a timestamp showing when the review was posted.

6. *Wish* - Also has a Wish ID(Primary Key), a Description (if the customer wants to add any general comments related to the Service he wants). There can be a specific Requirement period, (composite attribute) that contains Time, Day and Month when the customer wants to use the Service. He can also add a Maximum Price that he is willing to pay in return of the *Service* offered.
7. *Location* - To describe the location, following entities are used: *Street*, *City*, *State* and *Country*. They contain names of the respective entities.  
The following entities are used for the interaction between the Service Provider and the Customer:
8. *Question* is put up by the *Customer* to be answered subsequently by the *Service Provider*. It has a Question ID(Primary Key), Description and Timestamp (when the question was raised).
9. *Answer* is put up by the *Service Provider* in response to a *Question*. It has an Answer ID (Discriminating Key), Description (content of the Answer) and Timestamp.

### 3.2 Relationships

1. *Provides* - *Service Provider* provides a *Service* at a *Street* in a *City* in a *State* in a *Country*. It includes the Availability (a composite attribute containing Time, Day and Month) so that customers can book appointments, Description, Price (Cost incurred on the Service) and the Discount offered (if any). Note that a *Service Provider* can provide multiple services of the same type and different subtypes.
2. *Experience* - A *Customer* experiences a *Service* and puts a *Review* on it.
3. *Vote*, Up/Down vote by a *Customer* to a *Review*
4. *Wishes* - *Service* wanted by a *Customer*
5. *Lives*, a *Customer* lives in a *Street* in a *City* in a *State* in a *Country*
6. *Q&A*, *Question* raised by a *Customer* to a *Service Provider* and the subsequent *Answer*
7. *Friend*, A *Customer* is a friend of another *Customer*
8. *Message*, sent by a *Customer* to a *Customer*
9. *Bids* - A *Service Provider* offers to fulfil a *Customer's* Wish at a particular price, that is the Bid Value. It may also contain a small Description.
10. *Appointment* - A *Customer* books an appointment for a particular *Service* to be offered by a *Service Provider* at a Location alongwith the details of the Appointment that includes the Scheduled time and Cost of the *Service*.



## 4 Test Plan

### 4.1 Data

For our application, the data would be generated randomly by a program or online by crawling a website. To test our application, we will manually generate SQL queries corresponding to the above functionalities and test it against the data collected.

### 4.2 Functionalities (provided to the customer)

1. Looking out for a service at a particular location, with a particular availability - by putting filters on services. He can also see the reviews provided by other people for that service.
2. Select a service by a service provider, review it and earn points for the same - Review must be properly linked to the service provider and his service. There cannot be a review for a service that does not exist.
3. Asking questions to a service provider which are subsequently answered by the service provider.
4. Interacting with other customers via personal messages - Message must go to only the customer intended.
5. UpVote/DownVote someone else's review on the basis of his own experience - Now, we have to store who upvoted which answer, and not to allow multiple upvotes on same answer by same user.
6. The customer can view his friend list, and reviews provided by them.

7. Book an Appointment on the basis of the Availability of the Service - A service can be booked only if it is available for the time period required by the customer.

### 4.3 Functionalities (provided to the customer)

1. Put out details of the services provided by him. HE can also offer discounts.
2. The Service Provider can get to see the customers who are interested in taking up his service (through their wishlist) and the maximum price they can pay for it, therefore he can take selective decisions to increase his customer base.
3. Can answer questions posted to him by the customers.

The above features are offered by our application. For search/querying, we will also be adding the option to order results by availability / discounts / price etc. This can directly be handled by 'order by' option provided by SQL.

## 5 Future Extensions

Other than the above mentioned, we will aim to model the following aspects if time permits.

1. **Customisation based on kind of service** - Depending on the kind of service one provides, he might want to have a customised profile through which he can properly showcase the facilities available to him. But at present, we are keeping a general information area for the same.
2. **Authentication of Service Provider** - How to know whether a service provider actually exists, or is it fake profile?
3. **Authentication of Reviews** - Did the customer actually receive a service? Or is he providing fake reviews (Note that this is partly handled by the concept of customer rating which we are including in the project.)
4. **Followers of a Service Provider** - We have also thought of the concept of "following" a service provider, i.e. receive his updates, reviews etc. via notifications.
5. **Integration with Social Platforms** - We can integrate it with existing social networks like Twitter, Facebook etc.

## 6 Relational Model

### 6.1 For Entity Sets

Assuming E/R Model Approach for hierarchy relationship in Users (Customer, ServiceProvider):

1. User(UserID, LoginID, Password, FirstName, LastName, EmailID, Photograph, ContactNumber)
2. Customer(UserID, Gender, DOB, Age, CumulativePoints, Rank)
3. ServiceProvider(UserID, Rating, Webpage)
4. Service(ServiceID, Name, Type, SubType, MiniDescription)

5. **Many-to-Many Bad Relational Design -**  
Review(ReviewID, Content, Rating, TimeStamp, VotedByCustomerUserID, TypeOfVote)
6. Wish(WishID, Description, MaximumPrice);
7. WishTime(WishID, StartTime, EndTime);
8. Question(QuestionID, Description, TimeStamp);
9. Answer(QuestionID, AnswerID, Description, TimeStamp);
10. Country(CountryName);
11. State(CountryName, StateName);
12. City(CountryName, StateName, City);
13. **Many-to-One Bad Relational Design-**  
Street(CustomerUserID, StreetName, CityName, StateName, CountryName)

**Note:**

1. We have removed the CustomerID and ServiceProviderID, as they were redundant in providing unique identification to a type of user. In the E/R approach, we are anyway storing the UserID in Customer and ServiceProvider tables, and hence no need for another unique attribute.
2. We assume that there are no two end times for same start time, thus we haven't considered it as a primary key.
3. For the Many to Many relationship Votes from Review to Customer, we have added CustomerId as the attribute of Review, whereas there should be another relation relating Customer and Review for Votes. In this case, Review Information will be repeated and there can be an issue of consistency.
4. For the Many to One relationship Lives from Customer to Street, we have added the CustomerID to Street as an attribute. This will not only lead to repetition of information and hence wastage of resources, but also create another problem. We have a ternary relation from ServiceProvider and Service to Street.

Now we will have to map one location of a service given by a service provider to all tuples corresponding to every customer living at that location, which not only wastes space, but will also complicate search and other functions. So, in the relation for relationships, we have not considered this model, but the correct model which assumes that there is no CustomerID in Street table.

For **OO Approach**, we have 3 relations again:

1. CustomerUser(UserID, LoginID, Password, Name, EmailID, Photograph, ContactNumber, Gender, DOB, Age, CumulativePoints, Rank)
2. ServiceProviderUser(UserID, LoginID, Password, Name, EmailID, Photograph, ContactNumber, Rating, Webpage)
3. CustomerServiceProviderUser(UserID, LoginID, Password, Name, EmailID, Photograph, Contact-Number, Gender, DOB, Age, CumulativePoints, Rank, Rating, Webpage)

**Note:**

1. Here, there is no table for only User, because user has total participation in the "is a" relationship, and has at least one role as a Customer and/or a ServiceProvider. We note that in this case as well, separate CustomerID and ServiceProviderID are not required to distinguish entries. So, we have removed the two attributes from the relations.
2. Another point to note here is the problem associated with this structure. Suppose there is a User, who has only a customer profile. He starts offering a service and then he makes a service provider profile associated to the same account. In that case, we have to shift all his details from CustomerUser table to CustomerServiceProviderUser table. This might be a cumbersome task. Also, there is no authority that dictates that UserIDs cannot be same in the three tables.

## 6.2 For Relationship Sets

1. Friend(CustomerUserID1, CustomerUserID2)
2. Message(SenderCustomerUserID, ReceiverCustomerUserID, Content, Timestamp)
3. Experience(CustomerUserID, ReviewID, ServiceID)
4. Bids(ServiceProviderUserID, WishID, BidValue, Details)
5. Wishes(CustomerUserID, WishID, ServiceID)
6. Provides(ServiceProviderID, ServiceID, StreetName, CityName, StateName, CountryName, Description, Rate, Discount)
7. ProvideTime(ServiceProviderUserID, ServiceID, StartTime, EndTime)
8. Appointment(CustomerUserID, ServiceID, ServiceProviderUserID, StreetName, CityName, StateName, CountryName, Rate, StartTime, EndTime)
9. QAndA(CustomerUserID, ServiceProviderUserID, QuestionID, AnswerID)

## 6.3 Foreign Keys

1. CustomerUserID and ServiceProviderUserId in any relation are foreign keys into UserID in Users.
2. ServiceID in any relation is foreign key into ServiceID in Services.
3. ReviewID in Experience is foreign key into ReviewID in Reviews.
4. WishID in Wishes, Bids and WishTime are foreign keys into WishID in Wish.
5. QuestionID in Answer is foreign key into QuestionID in Question.
6. QuestionID, AnswerID in QAndA are foreign keys into QuestionID, AnswerID in Answer.
7. For location names, CountryName is a foreign key into Country, StateName is a foreign key into State and CityName is a foreign key into City.

## 6.4 Functional Dependencies

1. As is obvious, all super keys determine the entire tuple.
2. Functional dependencies for derived attributes:
  - (a) In User relation:  $\text{LoginID} \rightarrow \text{Password}$ ,  $\text{EmailID} \rightarrow \text{LoginID}$ ,  $\text{Email-ID} \rightarrow \text{UserID}$  (In fact, LoginID and emailID are candidate keys)
  - (b) In Customer relation:  $\text{DOB} \rightarrow \text{Age}$ ,  $\text{CumulativePoints} \rightarrow \text{Rank}$

## 6.5 Assertions

1. In case of Appointments, the Start Time and the End Time should be ahead in the future (that is later than the Current time).
2. At the time of booking, the time for appointment should be subset of the time in Availability.
3. Rating should be between 0 to 5.
4. The timestamps in case of Questions, Answers and Reviews should be those of the past.
5. Discount should be between 0-100
6. Dates, days and months should be apt.
7. DOB should be of the past.
8. Contact No. should be of 10 digits.
9. Password should be of minimum 8 digits (includes one symbol, one number, one alphabet)
10. Vote type should be +1/-1.
11. Gender should be Male/Female.

## 7 Screen Design

Instead of attaching screenshots, we are adding links to the webpages we have designed. We have hosted it on the cse server. We have designed the following five pages:

1. **Login Page** - <http://www.cse.iitb.ac.in/~astha/ConsumerConnect/index.html>  
Users can SignIn or Create an Account through this page. They will be directed to their home page after providing valid username and password.
2. **Customer's Home Page** - <http://www.cse.iitb.ac.in/~astha/ConsumerConnect/cons.html>  
This is the default page of a Customer. The left sidebar has options which will direct him to corresponding information (eg. On clicking Friends' Reviews, he can see the latest reviews provided by his friends.)
3. **Customer's Profile** - <http://www.cse.iitb.ac.in/~astha/ConsumerConnect/consprofile.html>  
This is the page when you other Customer's profile.



4. **Service Provider's HomePage** <http://www.cse.iitb.ac.in/~astha/ConsumerConnect/serviceprovider.html>

This is the default home page of a Service Provider. He can add more services, and search for peoples' wishes on this page.

5. **Service Provider's Profile** <http://www.cse.iitb.ac.in/~astha/ConsumerConnect/servprofile.html>

This is the page when a Customer visits the profile of a Service Provider.

## 8 Member Specific Roles

We haven't yet decided upon specific goals that will be assigned to each member. In general, we will all be helping out each other in accomplishing short term goals.