CIS 550 Project Milestone 2

Group Name: UltraDobe
Members: Shi Cheng, Zeyu Li, Wei Song, Tianchen Zhang

Motivations

1. Recommend restaurants to user according to his/her review history and current information such as GPS location and current time.
2. Provide user with filter criterion to make him/her be able to select restaurants according to his/her preference.
3. Allow user to invite either friends or strangers to have meal according to, but not limited to, the restaurant he/she chose and the stars other users ranked.

Features

Since Yelp provides a large amount of data of different restaurants as well as easy ways to access them, we decided to use them in our project. Inspired by Amazon online shopping mall, we would like to recommend restaurants to users according to a series of criterion such as ambience and their stars, which saves users' time to select among hundreds of restaurants, and help them quickly locate some excellent ones they might be interested in.

In addition to automatic recommendation system, we also provide users with a more flexible way to filter those restaurants according to their preferences. Among the criterion we offer, user is able to select part of them he/she wants to use in order to filter.
Inspired by the idea of social network, we decided to implement an interface that allows a user sends invitation to others so they have the chance to enjoy meals together. This would make our app a platform for people to make friends and expand their network rather than a simple restaurant-recommendation system.

Technology and tools

Ruby on rails, SQL, Google API, Git, AWS

Member responsibility

1. **Front end, web design**: Shi Cheng, Tianchen Zhang
2. **Google map implementation**: Zeyu Li
3. **Data base design and query**: Wei Song, Zeyu Li, Tianchen Zhang

Schema Desgin

As Figure 1 shows, there are 4 entities—Businesses, Restaurants, Users and Reviews, where Restaurant is an instance of Business.

The following are all the tables that are supposed to be create in the schema:

```
create table Businesses{
        type varchar(20),
        business_id Integer
        name varchar(20),
        full_address varchar(30),
        city varchar(20),
        state char(2),
        latitude varchar(8),
        longitude varchar(8),
        stars float,
```

```
        hours varchar(15),
        reviewNum integer,
        Primary key (business_id)
}
```

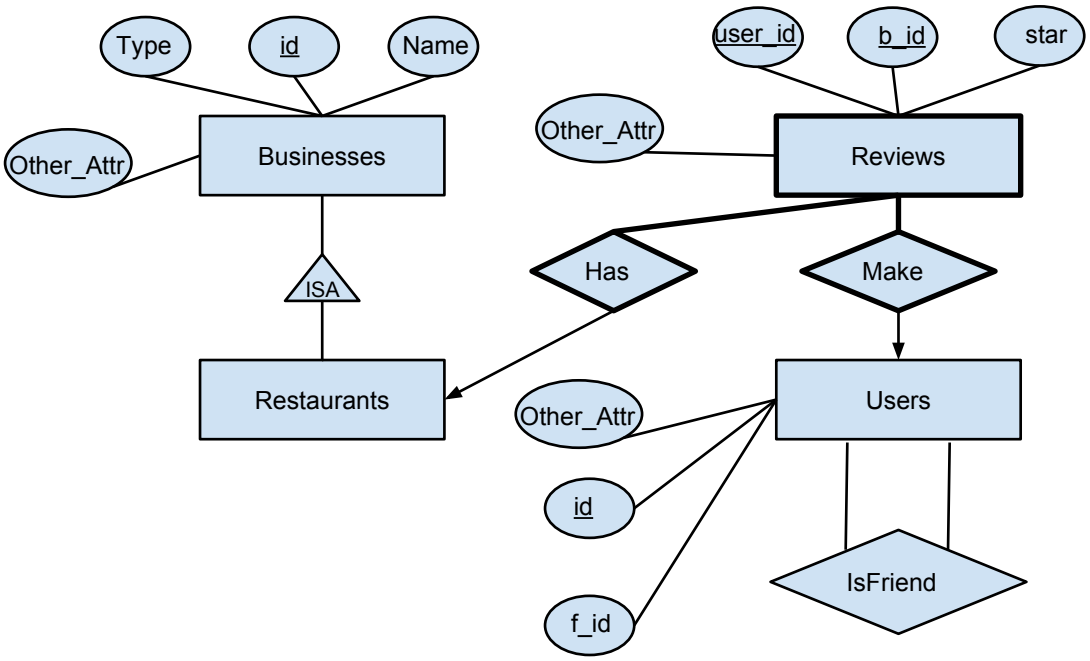

Figure 1

```
create table Restaurants{
        type varchar(20),
        rest_id integer
        name varchar(20),
        full_address varchar(30),
        city varchar(20),
        state char(2),
        latitude varchar(8),
        longitude varchar(8),
        stars float,
        hours varchar(15),
        reviewNum integer,
        Primary Key (rest_id)
        Foreign Key (rest_id) References Businesses
}

create table Reviews{
        type varchar(20),
        b_id integer,
        user_id integer,
        stars float,
        text varchar(100),
```

```
        date Date;
        Primary Key (b_id, user_id),
        Foreign Key (user_id) References Users,
        Foreign Key (b_id) References Businesses
}

create table Users{
        type varchar(20),
        user_id integer,
        name varchar(20),
        reviewNum integer,
        f_id integer
        averageStar float,
        Primary Key (user_id)
}

create table isFriend{ // relation
        friendOfMe integer,
        friendOfOther integer,
        Primary Key (friendOfMe, friend OfOther),
        Foreign Key (friendOfMe) References Users,
        Foreign Key (friendOfOthers) References Users
}
```