Project Outline (Milestone 2)

1. Motivation

We love food, we like making friends, and we care about our health. If only we could combine all those in one app - the NewBee Team App, which allows us to search restaurants by the criteria we specify, to find people who have interests in the same restaurants like we do, and to satisfy our endless curiosity to discover the relationship between health and different characteristics of restaurants.

2. List of Main Features

- Restaurant Recommender:
  - A query that joins the datasets photo and business and shows the relevant information and photos about the chosen business based on the user selected criteria (e.g. star, categories, location, etc.), sorted (ascending / descending) by the designated attributes.

- Friend Network:
  - Make friends & meet new people with someone you share the same taste. This function can get the user’s list of highest star reviewed business from a subquery of dataset user and review, and then provide the recommended n-connection other users that have high reviews for the same or related business, and provide a friend recommendation list for the user.

- Restaurant_Scientist: a series of analyses on restaurants and health
  - Best/Worst_business: A query that searches from dataset business to find the best/worst business, which has the most reviewed highest/lowest star value in a given area.
  - Distribution_of_business_by_star: A query that calculates the count and percentage of businesses in an area (aggregated by zipcode, or county, or state) by star level from the dataset business.
  - Relationship_health_price: A query that joins the datasets business and county health rankings, provide each area’s (county, or state) health ranking and price range frequency distribution.

3. List of more possible features

- Restaurant Recommender:
  - A query that joins the datasets photo, business, user and review. This query can get the user's list of highest star reviewed businesses from a subquery using the datasets user and review, and then provide a list of recommended other businesses that are similar to the subquery result in terms of categories, stars, etc.
- Restaurant_Scientist:
  - A query that searches from dataset business to find 5 star business sorted by number of reviews in a given area/of a given type.

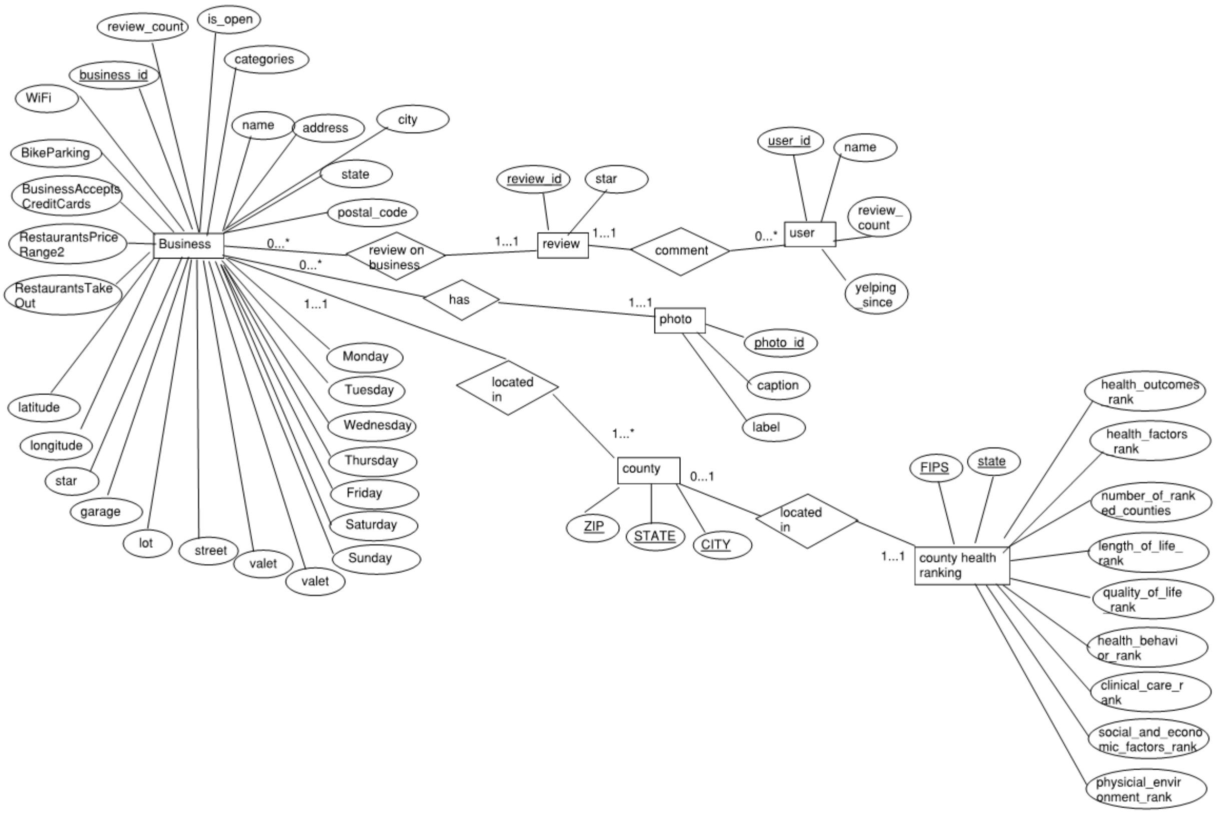
- A query that calculates the count and percentage of businesses in an area (aggregated by zipcode, or county, or state) by price level from the dataset business
- A query that calculates the count and percentage of businesses in an area (aggregated by zipcode, or county, or state) by restaurant type from the dataset business
- A query that calculates the average price level and review count in an area (aggregated by zipcode, or county, or state) from the dataset business
- A query that joins the datasets business and county health rankings, provide each area's (county, or state) health ranking and star rating frequency distribution
- A query that joins the datasets business and county health rankings, provide each area's (county, or state) health ranking and restaurant type frequency distribution

4. List of application pages

- Page 1: Restaurant Recommender. This page basically implements the search function Restaurant Recommender, which allows the user to input some parameters by which the search function will return a list of recommended restaurants which meet the criteria
- Page 2: Friend Network. This page shows restaurants that were reviewed by a given user and highly rated. Users can find other people who also left high-rate reviews on same restaurants. This page can show a list of recommended people with one up to three connections to users.
- Page 3: This page is the interesting one, which allows the user to discover the relationship between two datasets (i.e. businesses & county health rankings) on their own by specifying the attributes and seek the relationship between them.

5. ER diagram

The ER diagram for the relational database is as follows:



6. SQL DDL

the SQL DDL for dataset are listed as below:

6.1 Business

```
CREATE TABLE Business
(
  business_id            varchar(50),
  name                varchar(100),
  address               varchar(255),
  city               varchar(20),
  State               varchar(10),
  postal_code            varchar(20),
  latitude             float(20,5),
  longitude              float(20,5),
  stars              float(3,1),
  review_count          int,
  is_open             int,
  categories             varchar(255),
  WiFi               varchar(10),
  BikeParking            varchar(8),
  BusinessAcceptsCreditCards  varchar(8),
  RestaurantsPriceRange2      varchar(3),
  RestaurantsTakeOut       varchar(8),
  garage              varchar(8),
  lot                 varchar(8),
  street              varchar(8),
  valet               varchar(8),
  validated           varchar(8),
  Monday              varchar(16),
  Tuesday             varchar(16),
  Wednesday           varchar(16),
  Thursday            varchar(16),
  Friday              varchar(16),
  Saturday            varchar(16),
  Sunday              varchar(16),
  PRIMARY KEY (business_id)
);
```

6.2 user

```
CREATE TABLE user
(
  user_id        varchar(50),
  name           varchar(50),
  review_count   int,
  yelping_since  varchar(50),
  PRIMARY KEY (user_id)
);
```

6.3 review

```
CREATE TABLE review
(
  review_id          varchar(50),
  user_id            varchar(50),
  business_id        varchar(50),
  stars              float(3,1),
  PRIMARY KEY(review_id)
); #delete foreign key make the review can run faster in DDL
```

6.4 photo

```
CREATE TABLE photo
(
  photo_id           varchar(50),
  business_id        varchar(50),
  caption            varchar(255),
  label              varchar(10),
  PRIMARY KEY (photo_id),
  FOREIGN KEY (business_id) REFERENCES Business(business_id)
);
```

6.5 County_health_rankings

```
CREATE TABLE County_health_ranking (
        fips char(5),
        state varchar(20),
        county varchar(21),
        number_of_ranked_counties int,
        length_of_life_rank int,
        quality_of_life_rank int,
        health_behaviors_rank int,
        clinical_care_rank int,
        social_and_economic_factors_rank int,
        physical_environment_rank int,
        health_outcomes_rank int,
        health_factors_rank int,
        PRIMARY KEY(fips state));
```

6.6 Zip_county_crosswalk

```
CREATE TABLE Zip_county_crosswalk (
        zip char(5),
        state char(2),
        city varchar(27),
        fips char(5),
        PRIMARY KEY(zip, state, city),
        FOREIGN KEY (fips, state) REFERENCES County_health_ranking(fips, state));
```

7. Data clean pre-process

7.1 . Table: Yelp Business

The original data has 160585 businesses, and 14 arities as the table below. Most of the attribute could be useful for the project queries and shall remain in the data. But there are null values and the “attributes” and “hours” columns are dictionary type data, which include multiple values in one cell. The data need to be normalized to at least 1NF and refined to more useful form.

Original Table: Yelp_Business	
Size	124.4MB
Cardinality	160585
Arity	14
Attribute: business_id	varchar(22), e.g. ‘6iYb2HFDywm3zjuRg0shjw’
Attribute: name	varchar(40), e.g. ‘Oskar Blues Taproom’

Attribute: address	varchar(50), e.g. ‘921 Pearl St’
Attribute: city	varchar(30), e.g. ‘Boulder’
Attribute: state	varchar(3), e.g. ‘CO’
Attribute: postal_code	varchar(7), e.g. 80302
Attribute: latitude	float, e.g. 40.017544 (mean=38.76, std=7.13, min=27.998972, med=42.177366, max=49.490000)
Attribute: longitude	float, e.g. -105.283348 (mean=-94.27, std=19.98, min=-123.393929, med=-84.383281, max=71.113271)
Attribute: stars	float, e.g. 4.0 (mean=3.66, std=0.94, min=1.0, med=4.0, max=5.0)
Attribute: review_count	int, e.g. 86 (mean=51.96, std=130.03, min=5, med=17, max=9185)
Attribute: is_open	int, e.g. 1 (value_count={1: 123248, 0: 37337})
Attribute: attributes	JSON, e.g. {'RestaurantsTableService': 'True', 'WiFi': "u'free'", 'BikeParking': 'True', 'BusinessParking': "{'garage': False, 'street': True, 'validated': False, 'lot': False, 'valet': False}", 'BusinessAcceptsCreditCards': 'True', 'RestaurantsReservations': 'False', 'WheelchairAccessible': 'True', 'Caters': 'True', 'OutdoorSeating': 'True', 'RestaurantsGoodForGroups': 'True', 'HappyHour': 'True', 'BusinessAcceptsBitcoin': 'False', 'RestaurantsPriceRange2': '2', 'Ambience': "{'touristy': False, 'hipster': False, 'romantic': False, 'divey': False, 'intimate': False, 'trendy': False, 'upscale': False, 'classy': False, 'casual': True}", 'HasTV': 'True', 'Alcohol': "'beer_and_wine'", 'GoodForMeal': "{'dessert': False, 'latenight': False, 'lunch': False, 'dinner': False, 'brunch': False, 'breakfast': False}", 'DogsAllowed': 'False', 'RestaurantsTakeOut': 'True', 'NoiseLevel': "u'average'", 'RestaurantsAttire': "'casual'", 'RestaurantsDelivery': 'None'} * Some records have empty attributes in this

	instance
Attribute: categories	varchar(200), e.g. 'Gastropubs, Food, Beer Gardens, Restaurants, Bars, American (Traditional), Beer Bar, Nightlife, Breweries' * Some records have empty categories in this instance
Attribute: hours	JSON, e.g. {'Monday': '11:0-23:0', 'Tuesday': '11:0-23:0', 'Wednesday': '11:0-23:0', 'Thursday': '11:0-23:0', 'Friday': '11:0-23:0', 'Saturday': '11:0-23:0', 'Sunday': '11:0-23:0'} * Some records have empty hours in this instance

7.2 Table clean process for Yelp Business

- The most important attributes for the project are: address, postal_code, attributes, categories. Null check was performed for these attributes and the Null Ratio is about 9.3%. It is an acceptable amount to be dropped. The cleaned data has 118757 row left without all the null value.
- The dictionary type data of attributes was normalized to multiple columns. The original separation got 39 additional columns. Null Ratio of these columns were checked and reviewed against the requirement of proposal queries, several important attributes with null ratio less than 60% were remained in the data set, including: WiFi (Null Ratio=55.3%), BikeParking(Null Ratio=42.8%), BusinessParking(Null Ratio=30.3%), BusinessAcceptsCreditCards(Null Ratio=15.6%), RestaurantsPriceRange2 (Null Ratio=34.9%), RestaurantsTakeOut(Null Ratio=57.6%).
- The attribute: BusinessParking is a dictionary type data, it was divided to multiple columns: garage, lot, street, valet, validated.
- The dictionary type data of hours was normalized to multiple columns. New columns include: Monday,Tuesday ,Wednesday,Thursday,Friday, Saturday,Sunday
- The cleaned dataset contains 29 attributes, 118757 rows.
- the table is 3NF, primary key is the “business_id”.
- considering the dataset scale comprehensively with other dataset: business=(118757,29), photo=(200000,4), user=(2189457,4), review=(8635403,4), and 6GB photo package, which are too much for our project purpose. So further optimization was carried out, data from state MA and OR was selected for the project. the final dataset size are: business=(44782,29), photo=(61044,4), user=(835224,4), review=(3185478,4), and 1.71GB photo package.

The final results are as table below:

Cleaned Table: Yelp_Business	
size	11.8MB
number of rows	44782
number of attributes	29
Attributes	**Sample content**
business_id	‘6iYb2HFDywm3zjuRg0shjw’
name	‘Oskar Blues Taproom’
address	‘921 Pearl St’
city	‘Boulder’
State	‘CO’
postal_code	‘80302’
latitude	40.01754
longitude	-105.283
stars	4
review_count	86
is_open	1
categories	‘Gastropubs, Food, Beer Gardens, Restaurants, Bars, American (Traditional), Beer Bar, Nightlife, Breweries’
WiFi	‘u'free'’
BikeParking	‘TRUE’
BusinessAcceptsCreditCards	‘TRUE’
RestaurantsPriceRange2	3
RestaurantsTakeOut	‘TRUE’
garage	‘FALSE’

lot	‘FALSE’
street	‘TRUE’
valet	‘FALSE’
validated	‘FALSE’
Monday	‘11:0-23:0’
Tuesday	‘11:0-23:0’
Wednesday	‘11:0-23:0’
Thursday	‘11:0-23:0’
Friday	‘11:0-23:0’
Saturday	‘11:0-23:0’
Sunday	‘11:0-23:0’

7.3 . Table: Yelp review

The original review dataset is huge. Refer to the table for the original dataset summary. According to the proposed project’s function, several useless attributes were dropped. The columns remain for the dataset contain: review_id, user_id, business_id, stars, and only review related with state MA and OR was remaining.

Original Table: yelp review	
size	6.94GB
number of rows	8635403
number of attribute	9
attribute: starts	mean=3.73, std=1.45, min=1.0, 25%, 3.0, 50%= 4.0, 75%=5.0, max= 5.0
attribute: useful	mean=1.24, std=3.20, min=0, 25%=0, 50%=0, 75%=1,max=758
attribute:funny	mean=0.42, std=1.86, min=0, 25%=0, 50%=0, 75%=0, max=610
attribute: cool	mean=0.50, std=2.24, min=0, 25%=0, 50%=0, 75% =0, max=732

Cleaned Table: yelp review	
size	221MB
number of rows	3185478
number of attribute	4
Attributes	**Sample content**
review_id	'lWC-xP3rd6obsecCYsGZRg'
user_id	'ak0TdVmGKo4pwqdJSTLwWw'
business_id	'buF9druCkbuXLX526sGELQ'
stars	4

7.4 . Table: Yelp users
The original review dataset is huge. Refer to the table for the original dataset summary. According to the proposed project's function, several useless attributes were dropped. The columns remain for the dataset contain: user_id, name, review_count, yelping_since, and only user related with state MA and OR was remaining.

Original Table: yelp user	
size	3788.8MB
number of rows	2189457
number of attribute	22
attribute: average_stars	mean=3.65, std=1.15, min=1, 25%=3, 50%=3.88, 75% =4.55, max=5
attribute: review_count	mean=21.7, std=76, min=0, 25%=2, 50%=5, 75%=15,max=15686
attribute:useful	mean=38, std=535, min=0, 25%=1, 50%=3, 75%=1.3, max=20438

Cleaned Table: yelp user	
size	41.1MB
number of rows	835224
number of attribute	4
Attributes	**Sample content**
user_id	‘q_QQ5kBBwlCcbL1s4NVK3g’
name	‘Jane’
review_count	1220
yelping_since	3/14/2005 8:26:35 PM

7.5 Table: Yelp Photo
This table is well formed. A package of photos needs to be uploaded to the future server for related queries. only photos related with state MA and OR was remaining.

yelp Photo	
size	4.14MB
number of rows	61044
number of attribute	4
attribute: photo_id	unique=199999, freq=2
attribute: business_id	unique=39438, freq=493
attribute:caption	unique=86333, freq=88090
attribute: label	unique=5, freq=4000

7.6 Table: County Health Rankings Outcomes & Factors Rankings
The original table contains 8 attributes, but only 6 of them are helpful to our purpose. There are 51 observations with missing values(they are state level rows only for displaying purpose) and 61 counties not ranked. These observations were dropped.

County Health Rankings Outcomes & Factors Rankings	
size	119KB
number of rows	3193
number of attribute	8 (useful attributes for our purpose: 6)
attribute: FIPS	varchar(5), e.g. 56045 unique = 39488
attribute: state	varchar(20), e.g. New York unique=51
attribute: number_of_ranked_counties	int, e.g. 2 (mean=93.98, std=55.13, min=1, med=83, max=243)
attribute: health_outcomes_rank	int, e.g. 1 (mean=47.18, std=41.57, min=1, med=37, max=243)
attribute: health_factors_rank	int, e.g. 2 (mean=47.82, std=41.73, min=1, med=38, max=243)

7.7 Table: County Health Rankings Outcomes & Factors Rankings
The original table contains 16 attributes, but only 10 of them are helpful to our purpose. There are 51 observations with missing values(they are state level rows only for displaying purpose) and 61 counties not ranked. These observations were dropped.

County Health Rankings Outcomes & Factors SubRankings	
size	182KB
number of rows	3193
number of attribute	16 (useful attributes for our purpose: 10)
attribute: FIPS	varchar(5), e.g. 56045 unique = 39488
attribute: state	varchar(20), e.g. New York unique=51
attribute: number_of_ranked_counties	int, e.g. 2 (mean=93.98, std=55.13, min=1, med=83, max=243)
attribute: length_of_life_rank	int, e.g. 1 (mean=47.18, std=41.57, min=1, med=37, max=243)

attribute: quality_of_life_rank	int, e.g. 2 (mean=47.82, std=41.73, min=1, med=38, max=243)
attribute: health_behavior_rank	int, e.g. 3 (mean=47.82, std=41.73, min=1, med=38, max=243)
attribute: clinical_care_rank	int, e.g. 4 (mean=47.82, std=41.73, min=1, med=38, max=243)
attribute: social_and_economic_factors_rank	int, e.g. 5 (mean=47.82, std=41.73, min=1, med=38, max=243)
attribute: physicial_environment_rank	int, e.g. 6 (mean=47.82, std=41.73, min=1, med=38, max=243)

The two tables above both have FIPS which represent a unique county as primary key, so they can be combined and still be 3NF. Therefore, the final cleaned table is as follows:

Cleaned Table: County Health All Rankings	
size	222KB
number of rows	3081
number of attribute	12
Attributes	**Sample Contents**
fips	56045
state	New York
number_of_ranked_counties	99
length_of_life_rank	1
quality_of_life_rank	2
health_behavior_rank	3
clinical_care_rank	4
social_and_economic_factors_rank	5
physicial_environment_rank	6
health_outcomes_rank	7

health_factors_rank	8

7.8 ZIP to County Crosswalk

We realize that we need to have a crosswalk between zip, state, city and county in order to link the yelp dataset and county health rankings dataset. The dataset contains 8 attributes and only 4 of them are useful for our purpose. We also subset this dataset to the counties available in the county health rankings dataset. We also renamed the variables to make the variable names concise and consistent.

ZIP to County Crosswalk	
size	4MB
number of rows	54260
number of attribute	8 (useful attributes for our purpose: 4)
attribute: ZIP	varchar(5), e.g. 24012 unique = 39488
attribute: county	varchar(5), e.g. 16073 unique=3228
attribute: usps_zip_pref_city	varchar(27), e.g. San Francisco unique=18507
attribute: usps_zip_pref_state	varchar(2), e.g. ME unique=56

Cleaned Table: ZIP to County Crosswalk	
size	1.7MB
number of rows	53628
number of attribute	8 (useful attributes for our purpose: 4)
Attributes	**Sample content**
zip	varchar(5), e.g. 24012 unique = 39488
city	varchar(27), e.g. San Francisco unique=18507

state	varchar(2), e.g. ME unique=56
county	varchar(5), e.g. 16073 unique=3228

8. List of technologies

- AWS MySQL was used as the database, AWS S3 was used to store the photo data.
- SQL will be used to query the relational database.
- Exploratory analyses were conducted with Python programming.
- Both the client side and serve side programming will be implemented in JavaScript.
  - The serve side implementation will mainly use the Node.js framework.
  - The client side implementation will mainly use the React.js framework.

9. Group members' responsibility

- Jinjie Fan and Anqi Wang will be responsible for building the back-end of our application, including but not limited to creating the database and queries corresponding to the features that will be utilized by the front-end.
- Guihe Li and Yusheng Ding will be building the front-end of the application, including but not limited to structuring and designing the web pages, determining the balance between functionality and aesthetic appearance, etc.