**Brand name for our project：**

Terp Foodie

1. **restaurant processes**

We built a database for the restaurants in College Park to track each restaurant to analyze what kind of restaurants attract more customers’ reviews and higher ratings.

Each restaurant has a unique identifier and is required to restore restaurant name, address and restaurant categories.

Each Dish has a unique identifier and is required to restore the dish name, dish price, dish category, and dish review number.

Each platform has a unique platUrl and is required to restore the plat name, platFounder, platProduct, and platBusinessType.

Each user has a unique identifier and is required to restore the user name and user review count, number of useful and user start date .

1. **Identify at least four entity types.**

Restaurant, Dish, Platform, User

1. **Perform database analysis on entity and relationship types, i.e.**

**ER schema.**

Restaurant (**restaurantId,** restaurantName, restaurantCag, restaurantIsOpen, restaurantAddress, -restaurantStreet, -restaurantCity, -restaurantState, -restaurantZipCode, restaurantGeoCoord, -restaurantGeoLatitude, - restaurantGeoLongitude)

Dish (**dishId**, dishName, dishPrice, dishCategory, =dishReviewNum)

Platform (**platName**, platUrl, platFounder, platProduct, platBusinessType)

User (**userId**, userName, =userReviewCount, userStDate, userUseful)

**Relationship:**

**(***1. Identify relationship types including at least one many-to-many binary or*

*higher-degree relationship type - no singleton entity type.***)**

Produce: binary relationship

1 dish to 1 restaurant

1 restaurant to 0 or more dishes

Record: binary relationship

1 platform to 1 or more restaurants

1 restaurant to 1 or more platform

Review (reviewDate): binary relationship

1 user to 0 or more restaurants

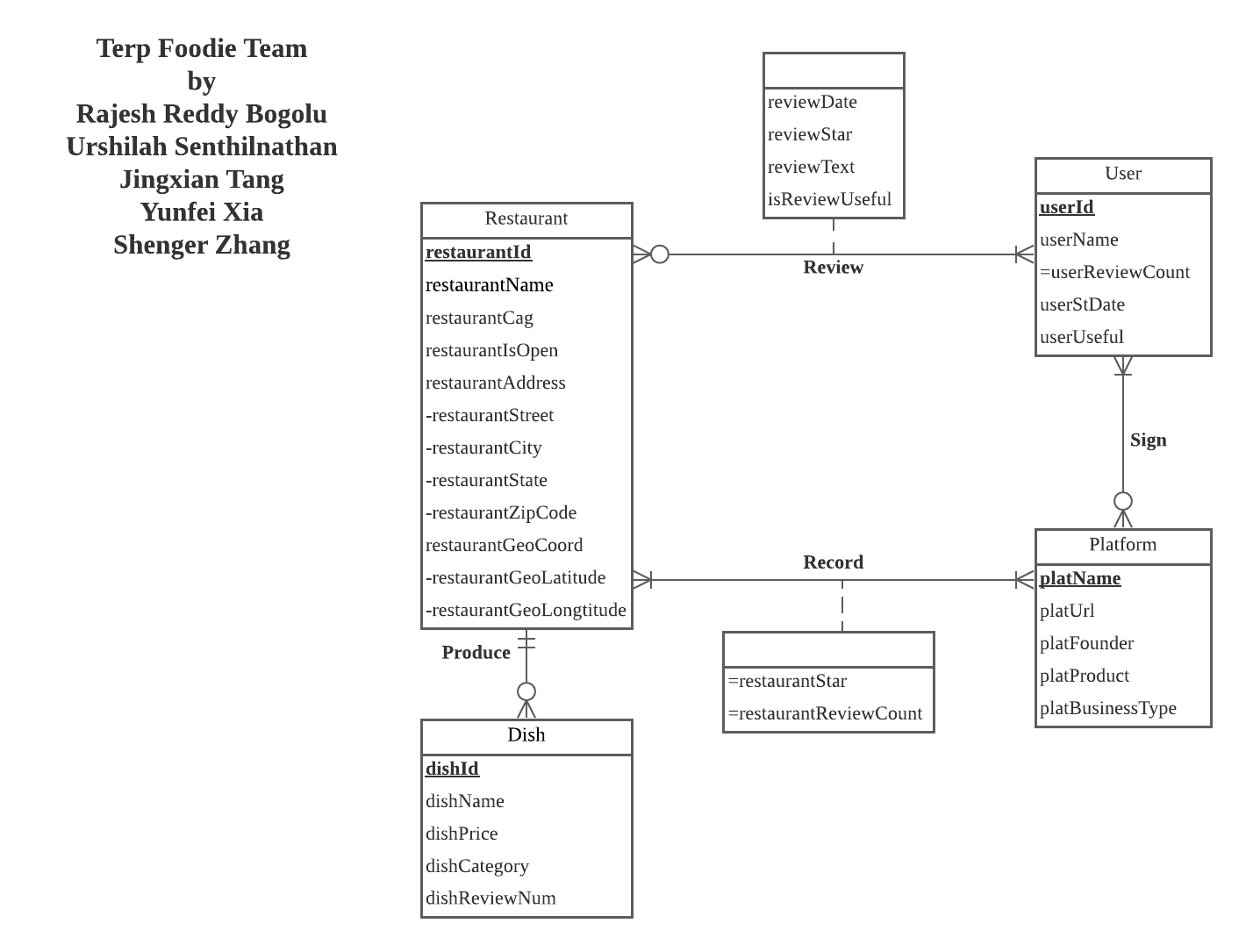
1 restaurant to 1 or more users

Sign: binary relationship

1 user to 0 or more platforms

1 plat to 1 or more users

1. **Design an ER diagram on Lucidchart – sharable to** [**adamlee@umd.edu**](mailto:adamlee@umd.edu)

****

1. **State mission statement(s) and mission objectives for the client.**

**A mission statement explains the company's purpose and focus, while objectives outline a path for achieving the mission.**

**Mission Statement :**

1. To connect people with great local restaurants along the way, Terp foodie hopes to enrich the lives of consumers and small business owners. In pursuit of this mission, we aspire to provide the most helpful information possible about local restaurants.
2. To analyze the customers’ reviews to help local restaurants elevate service quality, Terp foodie hopes to build insight on the food preference from the customers’ reviews and ratings.

**Mission objective :**

Using data analysis tools, we would find out the top-rated local businesses.

* To find the top 5 categories of local restaurants that get the highest rating.
* To find the top 5 categories of local restaurants that get the most customer reviews.
* To find the restaurant names and addresses of the restaurants that get the top 10 highest ratings.
* To find the restaurant names and addresses of the restaurants that get the top 10 most customer reviews.
* Using data analysis tools, we would analyze the customers’ affection for the restaurant.
* To find the top 20 dishes description and prices and categories that get the most customer reviews.
* To find the top 5 most useful reviews for the restaurant.

**Convert ER model into the relational schema and identify primary and foreign keys.**

Restaurant (**restaurantId***,*restaurantName, restaurantCag, isOpen, restaurantStreet, restaurantCity, restaurantState, restaurantZipcode, geoLatitude ,geoLongitude)

dishes (**dishId**, *restaurantId***,** dishName, dishPrice, dishCategory, dishReviewNum)

User (**userId,** userName, userStDate, userUseful)

Review(***userId***, ***restaurantId***, reviewDate,reviewStar, reviewText, isReviewUseful)

Platform (**platName**,platURL,platFounder,platProduct,platBusinessType)

Record (***restaurantId***, ***platName***, restaurantStar, restaurantReviewCount)

Sign (***platName,userId*)**

1. **Generate business rules and determine referential integrity actions.**

***Business rules:***

1. [R1] When the restaurant is closed or the information of the restaurant changes, the information of the restaurant’s dish should be deleted or changed.
2. [R2] When a restaurant is closed, the restaurant’s review should be deleted.
3. [R3] When the restaurant’s information is changed, the information on the review should be changed.
4. [R4] When a user logs out, the user’s review should still remain.
5. [R5] When a user’s information is changed, the user’s information on the review should also change.
6. [R6] When a restaurant is closed or the information of the restaurant changes, the information of the restaurant should be deleted or changed from the platform’s record.
7. [R7] When a platform is closed or the information of the platform changes, the platform’s information in the records should be deleted or changed.
8. [R8] When a user logs out, the user’s information should still remain in the platform
9. [R9] When users change their information, the user’s information should change in the platform
10. [R10] When a platform is closed or the information of the platform changes, the user cannot sign in to that platform.
11. [R11] When a platform changes its information like URL, the user can sign in to that new URL.

***Referential integrity:***

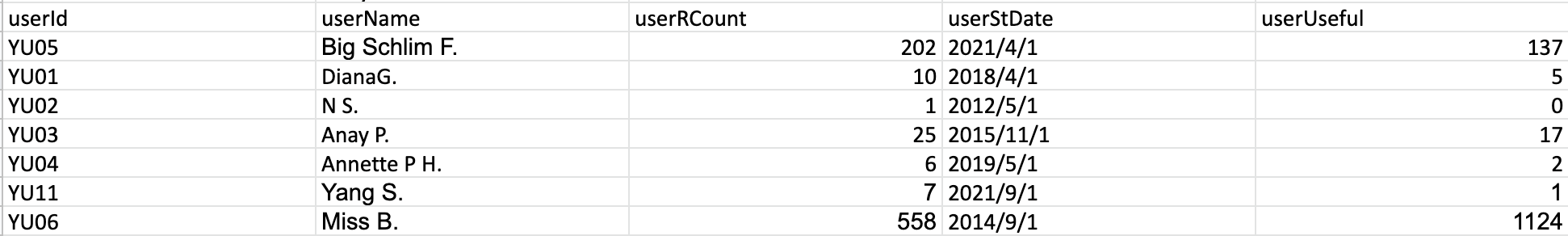
| Relation | Foreign Key | Base Relation | Primary Key | Business Rule | Constraint: ON DELETE | Business Rule | Constraint: ON UPDATE |
| --- | --- | --- | --- | --- | --- | --- | --- |
| dishes | restaurantId | Restaurant | restaurantId | R1 | CASCADE | R1 | CASCADE |
| Review | restaurantId | Restaurant | restaurantId | R2 | CASCADE | R3 | CASCADE |
| Review | userId | User | userId | R4 | NO ACTION | R5 | CASCADE |
| Record | restaurantId | Restaurant | restaurantId | R6 | CASCADE | R6 | CASCADE |
| Record | platName | Platform | platName | R7 | CASCADE | R7 | CASCADE |
| Sign | platName | Platform | platName | R10 | CASCADE | R11 | CASCADE |
| Sign | userId | User | userId | R8 | NO ACTION | R9 | CASCADE |

1. **Describe sample data for every relation.**

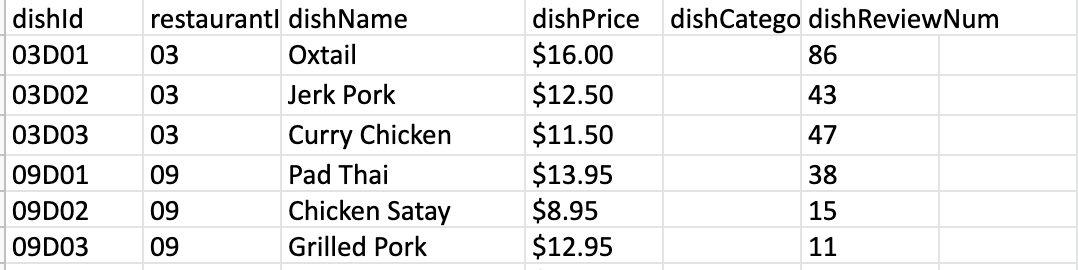
The restaurant’s entity will contain 20 restaurants from each platform. The category will contain at least 5 types to make the comparison more meaningful. The sample data will be like this.



The User entity will contain more than 10 users from each platform. The sample data will be like this.



The Dish entity will contain More than 5 dishes from each restaurant. The sample data will be like this.



The Reviews relation will contain 20 Latest reviews. The sample data will be like this.



The Platform entity will contain 5 platforms: Yelp, GoogleReview, TripAdvisor, Grubhub, OpenTable. The sample data will be like this.



The Record relation will contain approximately 100 relations. The sample data will be like this.



The Sign relation will contain more than 20 relations. The sample data will be like this

