# Project_0506_02_Proposal

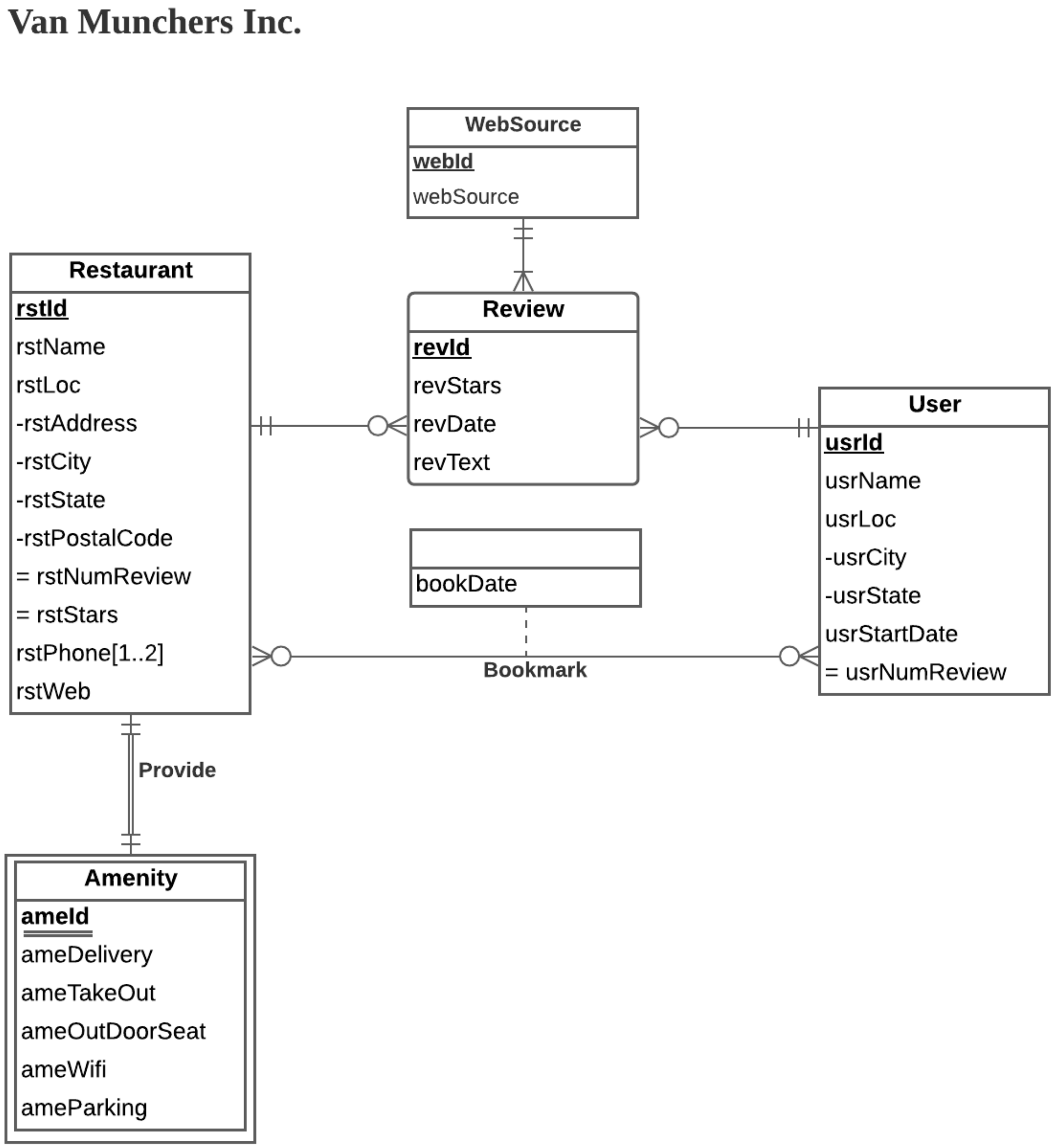
**Mission Statement:**

Our mission is to create a restaurants' database in College Park to enhance customers' experience and restaurants' services with detailed information.

**Mission Objectives:**

1. To find the top ten restaurants based on customers' ratings or numbers of reviews, tourists or new residents can try the best restaurants in College Park.
2. To assist residents or tourists to choose the restaurants with their priorities by offering restaurant amenities details.
3. To help restaurants to understand the customers' preferences more efficiently to improve their performance and profits.
4. To provide restaurants marketing recommendations by analyzing what websource the customers prefer and restaurants' reputation on the websource.
5. To understand the popularity of the restaurants among users by analyzing bookmarking trends.

**ER Diagram**:



**ER Schema**:

Restaurant (**rstId**, rstName, rstLoc, -rstAddress, -rstCity, -rstState, -rstPostalCode, =rstStars, =rstNumReview, rstPhone[1..2], rstWeb)

User (**usrId**, usrName, usrLoc, -usrCity, -usrState, usrStartDate, =usrNumReview)

Websource (**webId**, webSource)

Amenity (**ameId**, ameDelivery, ameTakeout, ameOutDoorSeat, ameWifi, ameParking)

**Relationships**:

Bookmark (bookData): binary relationship
1 user to 0 or more restaurants
1 restaurant to 0 or more users

Review (revId, revDate, revStars, revText): ternary relationship
1 user and 1 restaurant to 1 or more websources
1 user and 1 websource to 0 or more restaurants
1 restaurant and 1 websource to 0 or more users

Provide (): binary relationship
1 restaurant to 1 amenity
1 amenity to 1 restaurant

**Business processes/transactions**:

Each restaurant is described by a unique ID, name, location (consists of address, city, state, and postal code), number of reviews and stars of the restaurant as well as phone numbers and website.

Each review is described by a unique identifier, stars of the review, date of the review, and the text of the review.

Each web source is identified by its unique web id and web source.

Each web source can have zero to many reviews while each review can belong to one and only one web source.

Each amenity is described by a unique ID and whether or not it provides a delivery option, a takeout option, an outdoor seating, wi-fi, and a parking space.

Each restaurant is described by one and only one set of amenities, and each set of amenitie is identified by one and only one restaurant.

Each restaurant can have zero to many reviews whereas each review can only be identified by one restaurant.

Each user is described by a unique id, the name of the user, the location where the user is (consists of the city where the user is, the state where the user is), date of the review written by the user and number of reviews written by the user.

Each user can bookmark zero or many restaurants and each restaurant can be bookmarked by zero or many users.

Each user can write zero or many reviews, but each review can be identified by only one user.

**Relations**:

Restaurant (**rstId**, rstName, rstAddress, rstCity, rstState, rstPostalCode, rstWeb)

RestaurantPhone (*rstId*, **rstPhone**)

User (**usrId**, usrName, usrCity, usrState, usrStartDate)

Review (**revId**, revDate, revStars, revText, *rstId, userId, webId*)

Bookmark (***rstId***, ***usrId***, bookDate)

Websource (**webId**, webSource)

Amenity (**ameId**, ameDelivery, ameTakeout, ameOutDoorSeat, ameWifi, ameParking, *rstId*)

**Business rules:**

[R1] When a restaurant is deleted or updated from the database, the restaurant phone number should be deleted or changed on the database.

[R2] When a user is deleted from the database, the review will not be deleted but the user will become anonymous and the userid column will be null.

[R3] When a user is updated in the database, the review from the database must also get updated.

[R4] When a restaurant is deleted or updated in the database, the review about the restaurant should also be deleted or updated.

[R5] When a web source is deleted in the database, the review from the web source will not be deleted but the column of webid should be null.

[R6] When a web source is updated in the database, the review from the database should also be updated.

[R7] When a user is deleted or updated in the database, the bookmark of the user should also be deleted or updated.

[R8] When a restaurant is deleted or updated in the database, the bookmark of the restaurant should also be deleted or updated.

[R9] When a restaurant is deleted or updated in the database, the amenity of the restaurant should also be deleted or updated.

**Referential integrity:**

| Relation | Foreign Key | Base Relation | Primary Key | Business Rule | Constraint: ON DELETE | Business Rule | Constraint: ON UPDATE |
|---|---|---|---|---|---|---|---|
| Restaurant Phone | rstId | Restaurant | rstId | R1 | CASCADE | R1 | CASCADE |
| Review | usrId | User | usrId | R2 | SET NULL | R3 | CASCADE |
| Review | rstId | Restaurant | rstId | R4 | CASCADE | R4 | CASCADE |
| Review | webId | Websource | webId | R5 | SET NULL | R6 | CASCADE |
| Bookmark | usrId | User | usrId | R7 | CASCADE | R7 | CASCADE |
| Bookmark | rstId | Restaurant | rstId | R8 | CASCADE | R8 | CASCADE |
| Amenity | rstId | Restaurant | rstId | R9 | CASCADE | R9 | CASCADE |

**Sample Data:**

Restaurant (‘RST001’, ‘Saburo Ramen’, ‘8503 Baltimore Ave’, ‘College Park’, ‘MD’, ‘20740’, NULL)

RestaurantPhone (‘RST001’, ‘3012201635’)

User (‘USR001’, ‘Mariah S’, ‘Washington’, ‘DC’, ‘11/1/2013’)

Review (‘REV001’, ‘4/24/2021’, 4, ‘My first time trying Ramen was here. It was really really good. Since then one of the chefs left the establishment and the Ramen doesn't taste the same. The Ramen is still good but not as great as it was.’, ‘RST001’, ‘USR001’, ‘WEB01’)

Bookmark (‘RST001’, ‘USR001’, ‘11/1/2021’)

Websource (‘WEB01’, ‘Yelp’)

Amenity (‘AME001’, ’0’, ‘1’, ‘0’, ‘1’, ‘1’, ‘RST001’)