HW 4. Application Development

# Submission Instructions

* Include your name on each of your java files.
* Package your files in .zip format.
* Submit by 11:59PM 11/12 on Canvas.

# Assignment

*Use JDBC to create the data access layer for the restaurant review application. Use the CREATE TABLE statements from the HW2 solution:* [*http://goo.gl/PKJHtm*](http://goo.gl/PKJHtm)*. Each table will correspond to a simple java class and data access object (DAO) java class. Re-use the JDBC data access layer example we had in class for the blog application:* [*http://goo.gl/YwghQg*](http://goo.gl/YwghQg)*.*

You should implement the following:

* ConnectionManager.java
  + Re-use the file from the blog application example, and update with your MySQL instance properties.
* Users.java/UsersDao.java
  + DAO methods:
  + public Users create(Users user)
  + public Users getUserByUserName(String userName)
  + public Users delete(Users user)
* CreditCards.java/CreditCardsDao.java
  + DAO methods:
  + public CreditCards create(CreditCards creditCard)
  + public CreditCards getCreditCardByCardNumber(long cardNumber)
  + public List<CreditCards> getCreditCardsByUserName(String userName)
  + public CreditCards updateExpiration(CreditCards creditCard, Date newExpiration)
  + public CreditCards delete(CreditCards creditCard)
* Companies.java/CompaniesDao.java
  + DAO methods:
  + public Companies create(Companies company)
  + public Companies getCompanyByCompanyName(String companyName)
  + public Companies updateAbout(Companies company, String newAbout)
  + public Companies delete(Companies company)
* Restaurants.java/RestaurantsDao.java
  + DAO methods:
  + public Restaurants create(Restaurants restaurant)
  + public Restaurants getRestaurantById(int restaurantId)
  + public List<Restaurants> getRestaurantsByCuisine(Restaurants.CuisineType cuisine)  
    Note: Define the CuisineType enum in Restuarants.java (see BlogUsers.java for an example).
  + public List<Restaurants> getRestaurantsByCompanyName(String companyName)
  + public Restaurants delete(Restaurants restaurant)
* SitDownRestaurants.java/SitDownRestaurantsDao.java
  + DAO methods:
  + public SitDownRestaurants create(SitDownRestaurants sitDownRestaurant)
  + public SitDownRestaurants getSitDownRestaurantById(int sitDownRestaurantId)
  + public List<SitDownRestaurants> getSitDownRestaurantsByCompanyName(String companyName)
  + public SitDownRestaurants delete(SitDownRestaurants sitDownRestaurant)
* TakeOutRestaurants.java/TakeOutRestaurantsDao.java
  + DAO methods:
  + public TakeOutRestaurants create(TakeOutRestaurants takeOutRestaurant)
  + public TakeOutRestaurants getTakeOutRestaurantById(int takeOutRestaurantId)
  + public List<TakeOutRestaurants> getTakeOutRestaurantsByCompanyName(String companyName)
  + public TakeOutRestaurants delete(TakeOutRestaurants takeOutRestaurant)
* FoodCartRestaurants.java/FoodCartRestaurantsDao.java
  + DAO methods:
  + public FoodCartRestaurants create(FoodCartRestaurants foodCartRestaurant)
  + public FoodCartRestaurants getFoodCartRestaurantById(int foodCartRestaurantId)
  + public List<FoodCartRestaurants> getFoodCartRestaurantsByCompanyName(String companyName)
  + public FoodCartRestaurants delete(FoodCartRestaurants foodCartRestaurant)
* Reviews.java/ReviewsDao.java
  + DAO methods:
  + public Reviews create(Reviews review)
  + public Reviews getReviewById(int reviewId)
  + public List<Reviews> getReviewsByUserName(String userName)
  + public List<Reviews> getReviewsByRestaurantId(int restaurantId)
  + public Reviews delete(Reviews review)
* Recommendations.java/RecommendationsDao.java
  + DAO methods:
  + public Recommendations create(Recommendations recommendation)
  + public Recommendations getRecommendationById(int recommendationId)
  + public List<Recommendations> getRecommendationsByUserName(String userName)
  + public List<Recommendations> getRecommendationsByRestaurantId(int restaurantId)
  + public Recommendations delete(Recommendations recommendation)
* Reservations.java/ReservationsDao.java
  + DAO methods:
  + public Reservations create(Reservations reservation)
  + public Reservations getReservationById(int reservationId)
  + public List<Reservations> getReservationsByUserName(String userName)
  + public List<Reservations> getReservationsBySitDownRestaurantId(int sitDownRestaurantId)
  + public Reservations delete(Reservations reservation)
* Inserter.java
  + For each class above: exercise the create, read, update, and delete operations (if the method exists) using the data access object.