CSC207/B07 Introduction to Software Design
Fall 2013 — Assignment 1

1 Logistics

- **Due date:** 10:00pm Thursday 10 October 2013
- **Group size:** Individual

2 Overview

For Assignment 1, you will implement a set of Java classes according to the UML diagrams provided in this assignment description. You will also provide unit tests for a subset of these classes.

3 Learning Goals

By the end of this assignment, you should have:

1. gained better understanding of specifying object oriented program design with UML
2. implemented a given OO design specified by a UML class diagram in Java
3. become more familiar with Java interfaces, abstract types, and inheritance
4. practiced designing and developing unit tests

4 What to do for this assignment

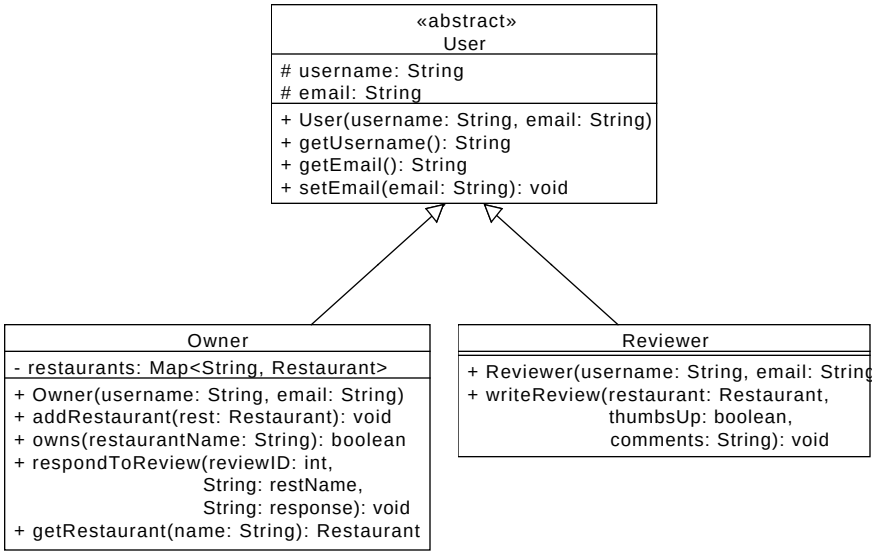
1. Write a set of Java classes that obey the specifications below. They must be in project `A1soln`, in package `a1soln`.
2. To submit your work, commit `A1soln`, `src`, `a1soln`, and your Java files under the existing directory `A1`.

   Do **not** commit the files and directories generated by Eclipse, such as `bin`, `.project`, etc.

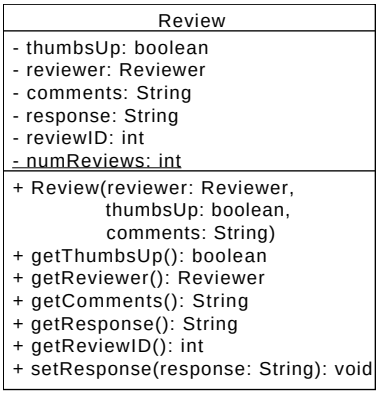
5 Restaurant Review System

Members of your software development team have designed a software system to facilitate restaurant reviews. Implement the Java classes with appropriate fields and methods according to the designs shown in the following diagrams and descriptions.

The system will have two types of `User`s: `Reviewer`s and `Owner`s. `Reviewer`s write reviews for restaurants, which involves writing comments and giving the restaurant a "thumbs up" or not. `Owner`s own `Restaurant`s and are able to write responses to `Review`s of their `Restaurant`s. The `Owner`s keep track of the `Restaurant`s that they own using a `Map` of restaurant name to `Restaurant`. This diagram outlines the system's `User` classes:



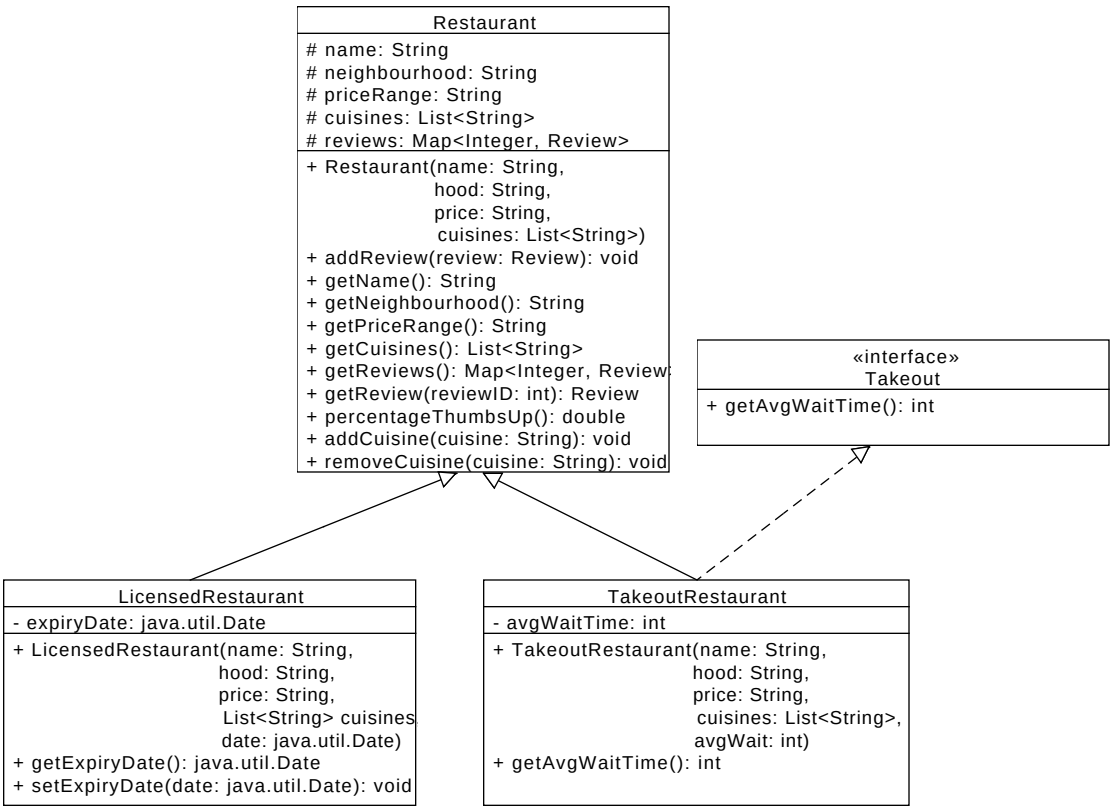
Each `Review` has a unique review ID, with the first reviewing have review ID 1, the second having review ID 2, and so on. There is a `static` variable used to keep track of the number of `Review`s and the review IDs are assigned based on the value of that `static` variable. This diagram outlines the system's `Review` class:



The constructor in `Review` should set the default value of `response` to `null`.

There are several types of restaurants: `Restaurant`s, `LicensedRestaurant`s and `TakeoutRestaurant`s. You may assume that the restaurant names are all unique. The `percentageThumbsUp` method returns the percentage of `Review`s for that `Restaurant` for which the `Reviewer` gave a thumbs up, as a value between `0.0` and `100.0`. The `Date` used by `LicensedRestaurant` is an instance of the built in class `java.util.Date`:

`http://docs.oracle.com/javase/7/docs/api/java/util/Date.html`



6 Testing

Use `JUnit` to thoroughly test your classes `Review` and `Owner`.

7 Evaluation

In addition to correctness, your submission will also be marked for style. Please follow the style guidelines outlined in `style.pdf`. Additionally, your choice of test cases will be evaluated based on guidelines outlined during lecture.

8 Checklist

Have you...

- run the type checker?
- followed the style guidelines?
- tested your code on the lab computers?
- committed all of your files (including `TestReview.java` and `TestOwner.java`) in the correct directory?
- verified that your changes were committed using `svn list` and `svn status`?