NYU Tandon School of Engineering
Computer Science and Engineering
CS 6083, Fall 2016

November 9, 2016

Project #1 (due November 25)

Your task in this semester's project is to design a website called *cookzilla*™ that focuses on cooking and recipes. The site should allow people to post cooking recipes, to review and grade posted cooking recipes, to attach additional suggestions to a posted recipe, to organize cooking meetings with other users, and possibly other mechanisms that you think will be useful. In this first part of the project, you have to design a relational database that can serve as a relational backend for this new web site. In the second part, you will then design the actual web site, which accesses this database.

Here are some more details about the idea behind this new site. Registered users on the site are identified by a unique user name, and they can optionally upload a short profile with information about themselves. Users can post recipes on the site that are then visible by everybody. A recipe usually has a title (e.g., "Aunt Mary's Apple Pie"), a number of servings, the ingredients and their quantities, a textual description of how to make the food item, and one or more pictures of the dish. Recipes can also link to other related recipes. The system also has a set of predefined tags, supplied by the website, to help organize recipes into categories. Examples of such tags would be "italian'', "chinese", "vegan", "soup", "spicy", etc. When a user posts a recipe, they can attach one or more of these tags to their recipe. Think a little about how to specify quantities for ingredients, since in recipes, people may use expressions such as ``200 grams of flour", "one ounce of water", "one teaspoon of Worcester Sauce", or "a pinch of salt". Note that these units can be converted, e.g., one teaspoon is about 5ml, and you may need to model such conversions.

Other users can read the posted recipes, can write reviews of the recipes, and can give a rating. A review has a title, a text for the review, and optionally some photos (since people who used the recipes may want to post pictures of the resulting dish) and some suggestions on how to modify the recipe (e.g., "use butter instead of oil" or "bake at slightly lower temperature for a more moist cake"). Ratings are from 1 to 5 stars. Finally, users can create and join informal groups that organize cooking events. Thus, there might be a group called ``Brooklyn Turkish Cooking" whose members occasionally meet to try out recipes for Turkish food. Members can RSVP for such meetings, and may post meeting reports with photos of the event to the site, where they can be read by members of the group but not by other users.

Both parts of the project may be done individually or in teams of two students. However, you have to decide on a partner and email TA Jubin Soni by Tuesday, 11/15. If we do not hear from you by that date, we will assume that you are doing an individual project. The second part of the project will be due before the final exam. In this first part of the project, you have to design the database backend of the system, that is, an appropriate relational schema with keys, constraints, and maybe views, plus a set of useful queries. You should use your own database system on your laptop or on an internet-accessible server. Use a system that supports text operators such as "like" and "contains", since you should allow users to search the content by keywords.

Note that the second project builds on top of this one, so you cannot skip this project. The system described here may of course have many similarities with existing cooking sites. Before starting your work, you should briefly think about how you envision the final system to look like at the end of the second part of the project, and what kind of operations and steps there are. For example, there should be a login page, a page where a user can sign up for the first time, and a page where users can create or update their profiles. There should be ways for users to post recipes and reviews, to rate recipes and add suggestions, to create and join groups and to organize and RSVP for cooking meetings. Users should also be able to search for recipes by keywords, tags, and ratings, and for extra credit you may add a

recommender system that suggests to users recipes they might like (say, based on their past likes, and on the likes of people that have similar tastes or are in many of the same groups). Some of the details are only important in the second part, but start thinking about them now.

Project Steps: Following is a list of suggested steps for this part of the project. Note that in this first part, you will only deal with the database side of this project - a suitable web interface will be designed in the second project, where you will also have a little more freedom in what to exactly implement (so that solutions may differ from each other more).

(a) Design, justify, and create an appropriate relational schema for the above situation. Make sure your schema is space efficient and suitably normalized. Show an ER diagram of your design, and a translation into relational format. Identify keys and foreign key constraints. Note that you may have to revisit your design if it turns out later that the design is not suitable for some of the queries or functionality. Provide a detailed explanation of your design decisions!

(b) Create the database schema, together with key, foreign key, and other constraints.

(c) Write SQL queries (or sequences of SQL queries) for the following tasks:

- Create a record for a new user account, with a name, a login name, and a password.
- List all recipes with tag “italian” that contain the keyword ``broccoli''.
- List all members of the group “Park Slope Cake Club” that have given a positive RSVP to more that three events of the group.
- List all recipes with tag “cake” that contain more than 50 grams of sugar per serving.
- Add a review with title “Yummy!”, text “Really, really, tasty!”, and a rating of 5 stars to the recipe for “Grandma’s Fettuccini Alfredo”.
- List all recipes containing the word “tuna”, sorted from highest to lowest average rating.
- List all recipes that are related to a recipe that contains the word “tuna”.

(d) Populate your database with some sample data, and test the queries you have written in part (c). Make sure to input interesting and meaningful data and to test a number of cases. Limit yourself to a few entries each, but make sure there is enough data to generate interesting test cases. It is suggested that you design your test data very carefully. Show your test data as tables, not as long lists of insert statements, and discuss the structure of the data. Print out and submit your testing.

(e) Document and log your design and testing appropriately. Submit a properly documented description and justification of your entire design, including ER diagrams, tables, constraints, queries, procedures, and tests on sample data, and a few pages of description. This should be a paper of say 10-15 pages with introduction, explanations, diagrams, etc., that you will then revise and expand in the second part.