EECS3421 Summer 2020
Assignment 3
Entity Relationship Diagram and Database Design

Due: Sat, Jun 20, 2020 at 11:59pm

Introduction

In this assignment, the main goal is to design an Entity Relationship Diagram and a database schema for an online Couchsurfing service/system. The general description of the system, as well as, its functional requirements are given in the following subsections. You will need to analyze the domain and prepare a report that contains a detailed specification of how information will be represented and stored in a database management system.

About Couchsurfing

Couchsurfing[1] is a platform for members to "surf" on couches of other members by staying as a guest at their home. The system offers its members, usually referred to as "couch surfers", hospitality exchange in terms of free couch to sleep on, showing around the place, information about local events organized by members, language exchange, helping each other with travel issues, etc. Basically, it is an online social network that connects users towards a global community of travelers. The objective of the system is to assist people to share their lives and experiences with the people they encounter, facilitate cultural exchange and bring mutual respect, hospitality, and friendship. It is reported that one of the most popular Couchsurfing systems has more than 12 million member profiles from 200,000 cities across the world.

System Functional Requirements

There are three user roles that interact in a Couchsurfing system:

- Guest (or surfer), who travels to other cities and stays with a host. They need to plan their trip and find a host whose place is available.
- Host, who receive couch requests from the guests and can accept/decline to host them for the duration mentioned in the requests.
- Administrator, who has administrative (moderator) rights of the site. They receive reports of potentially inappropriate groups (see about groups later) created by members. They review and examine the content of the reported group and can remove it if it is a commercial spam, threatening or abusive.

Guests and hosts should first sign up into the system. To do that, they need to provide some information such as email, gender, birth date, city and country of residence. A user can register multiple cities and countries of residence. Once signed up, they can provide further information about their profile, such as real name, interests, having/allowing pets, music, movies, books, etc. Moreover, each user can show if (s)he has a couch available to be surfed on by others. They can also define the time periods in which their couch is available. The system provides the following set of features that should be included in your design:

---

[1] See the Wikipedia article: http://en.wikipedia.org/wiki/CouchSurfing

- *Couch requests*: Users can send/receive couch requests. Each couch request includes data of arrival and departure dates, number of surfers, and status. The status could get any of the predefined set of values: accepted, pending, declined, or maybe. As part of the request, the sender can spend a few words describing his/her personality, reason of traveling, etc. By accepting a couch request, the host agrees to provide accommodation to the guest for the duration of the request. During/after the trip, the guest has the option of writing testimonial(s) for the host. A testimonial is a guest's written statement commenting about the hospitality of the host. Moreover, the guest can rate of the host's hospitality using a numerical rating scale that ranges between 1 (low) and 5 (high). The system uses these rates to calculate the overall credibility score of each member and presents that on their profile.
- *Friendships*: The system allows users to send friend requests to other members. Within the request, the sender shows the date they have met, friendship type, Couchsurfing experience they have had together (either as host or guest). Also, within the request, the sender decides either to show or hide the other party on his/her Friends List. Once the request is accepted, the friendship connection is held between two members.
- *Events*: Members can create and organize events. To create an event, the organizer needs to provide the date, time, location, and description of the event. Once an event is created, other members can join the event. Each member who is attending an event, can send invitation to other members.
- *Groups*: Members can create group(s). Each group has a description, type (public or private) and a category to which it belongs (e.g., places, people). Other members can join group(s). Each member of a group can send post(s) to the group. Posts are visible to all members of the group. Members of a group can reply to a given post. Response are visible to the other members of the group.
- *Search*: The users can search for available couches in a given city/country and for a specific time period. The user can filter the search results by selecting preferred gender of the host. The search engine returns a set of profiles that have a couch matching the search criteria. The system keeps the search history of the user. If a search returns no result (i.e., no available couch for that period of time and location), the system sends notification as soon as a couch becomes available.

Deliverable

The deliverable of this assignment is a report that should include the following parts:
- **Part A [10%]**: A detailed description of the assumptions you made about the system's functionality (minimum 10 assumptions, maximum 1 page).
- **Part B [40%]**: An Entity Relationship Diagram (ERD) of the system. Do not include any *isa* relationships or *weak entities* (these should be restructured to a final ERD). Your diagram should be **computer-generated** and **not** handwritten AND/OR scanned. You can choose to follow any of the notation conventions used in the slides but need to be consistent throughout.
- **Part C [40%]**: A translation of your ERD to a Relational Schema.
- **Part D [10%]**: A definition of your Relational Schema following the `PostgreSQL` syntax.

Submission Instructions

Your assignment must be typed; handwritten assignments will **not** be marked. You may use any word-processing software you like. Many academics use LaTeX. It produces beautifully typeset text and handles mathematical notation well. Whatever you choose to use, you need to produce a final document in *pdf*. On the first page of your pdf you should declare:

- your team (whether it is a team of one or two students) including login, first name, last name and student number.
- the number of any grace days you have used so far.

Important Notes
- If you are working in a pair, only one of you should submit.
- You should submit your document both electronically and as a hard copy in class.

Step 1: Electronic Submission

First, you should submit your work electronically using the submit command in PRISM lab computers. For this assignment, you will hand in just one file named **a3.pdf**. When you have completed the assignment, move your **a3.pdf** in a directory (e.g., assignment3), and use the following command to electronically submit your files within that directory:

```
% submit 3421 a3 a3.pdf
```

You may submit your solution as many times as you wish prior to the submission deadline. Make sure you name your file exactly as stated (including lower/upper case letters). You may check the status of your submission using the command:

```
% submit -l 3421 a3
```

Once you have submitted, be sure to check that you have submitted the correct version; new or missing files will not be accepted after the due date.