

**CSCE 5350 Fall 2010: Social Network Database Design**  
**Total: 350 points, Extra Credit: 60 points**  
**Due on: Pre-finals week (12/6/10 – 12/10/10)**

---

Social networks map and measure the connections and the flow of communication between individuals, and have become a major tool of interaction among users.

In this project, you will be developing a social networking database system based on popular real-world social networks such as Facebook, MySpace, Google Buzz, LinkedIn, etc., you can also explore networks with slightly uneven social structures such as Twitter (followed by, following users). While each network has its own unique features and different approaches to managing users' profiles, connections and privacy issues, their key features and basic structure are fairly consistent. You can either model your project on whichever real-world social network you are most familiar with, or you can be creative and design something a bit different from the existing ones (by adding nifty features you wish might have been available but aren't currently, or something you think could have been done/handled in a better way by your favourite social network).

Up to 3 students can team up for the project and will receive the same grade. For meeting the project requirements, you only need to develop command-line interfaces, but you will receive extra credit if you develop web interfaces.

Login procedure:

We will be using the MySQL server on the department's "csp" machines. For getting an account on the MySQL server, you need to go to <http://jsp.csci.unt.edu/> and follow the instructions on the page. It would probably take a day for your account to get created (although not necessarily). You can then login in to the MySQL server from any csp machines by using the command:

```
mysql -h richese -u userid -p
```

where "userid" would be replaced with your userid (or euid). The default password would be the same as your userid. You can change your password after your first login using the command:

```
SET PASSWORD FOR 'userid'@'%' = OLD_PASSWORD('newpass');
```

where "userid" would be replaced by your userid and "newpass" would be replaced by your new password.

All the above and a lot more information is available on the cse-esupport website here:

<http://esupport.cse.unt.edu/index.php?doc=prog-mysql-000>

A database with your userid as the database name would be created by default, in which you can create and populate tables. For accessing the database from your program through a command-line interface, you can either use Java or C/C++ with JDBC or ODBC/Pro\*C connection respectively. For accessing the database from your program through a web interface, you can use a scripting language like PHP or Perl. The csp machines support only PHP and Java/C/C++, so if you are using them, you will need to write and

store your scripts or programs on the csp machines and then insert code in your program to connect to the MySQL database. More information about PHP scripting is available here:

<http://esupport.cse.unt.edu/index.php?doc=prog-php-001>

There is no restriction on the languages you can use in the project. You are also free to use any other language that you might know (.NET framework, C#, etc.) to interface with MySQL, in which case you would have to use your own machines to develop the code.

### **Requirements:**

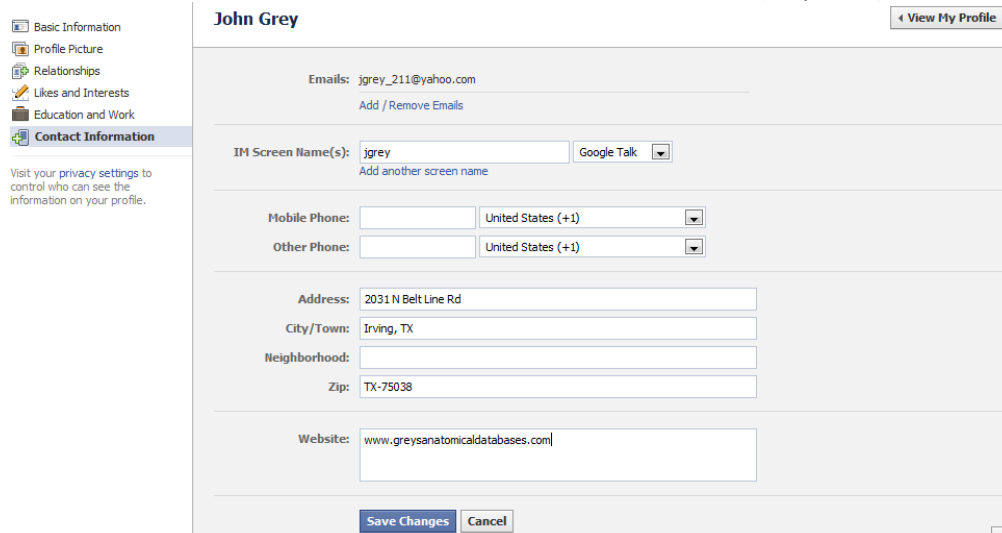
Following is a brief description (using Facebook as an example) of the requirements the system needs to meet. If you plan on developing web interfaces (although you do not have to do so), there are a few sample interfaces provided below. The basic idea remains the same, whether the interface is command-line or web-based. Note that the interfaces below are only meant to give a general idea of the requirements and you do not have to implement the interfaces exactly as they are shown below. You can take creative liberties, as long as the purpose of the interface remains the same.

1. Users should be able to log on with a userID or email address and password, or sign up if they are a new user: (30 points)



The screenshot shows a login and sign-up interface. At the top, there are fields for 'Email' (containing 'jgrey\_211@yahoo.com') and 'Password' (with masked characters). Below these are links for 'Keep me logged in' and 'Forgot your password?'. A 'Login' button is on the right. Below the login section is a 'Sign Up' section with the text 'It's free, and always will be.' The sign-up form includes fields for 'First Name' (John), 'Last Name' (Grey), 'Your Email' (jgrey\_211@yahoo.com), 'Re-enter Email', and 'New Password'. There is also a 'Select Sex' dropdown and a 'Birthday' section with 'Month', 'Day', and 'Year' dropdowns. A 'Sign Up' button is at the bottom.

2. Once logged in, users should be able to edit their profiles, contact info., and enter/edit other information such as likes, dislikes, interests, education, work, etc. : (30 points)



The screenshot shows a profile edit page for 'John Grey'. On the left is a navigation menu with options: 'Basic Information', 'Profile Picture', 'Relationships', 'Likes and Interests', 'Education and Work', and 'Contact Information' (which is highlighted). Below the menu is a link to 'Visit your privacy settings to control who can see the information on your profile.' The main content area is titled 'John Grey' and has a 'View My Profile' link. It contains several sections: 'Emails' (jgrey\_211@yahoo.com), 'IM Screen Name(s)' (jgrey), 'Mobile Phone' and 'Other Phone' (both with country dropdowns set to 'United States (+1)'), 'Address' (2031 N Belt Line Rd, Irving, TX, TX-75038), and 'Website' (www.greysanatomicaldatabases.com). 'Save Changes' and 'Cancel' buttons are at the bottom.

- Users should be able to send, receive friend requests and accept/ignore/decline them. Users should also be able to look up, join and un-join communities: (30 points)

[s](#)
[Mobile](#)
[Language](#)
[Payments](#)
[Facebook Ads](#)

each based around a workplace, region, high school, or college. Join a network to discover the people

**Join a Network**

**Enter a workplace or school.**  
 Network name:

**University of North Carolina School of the Arts**  
 Winston Salem, North Carolina

**The University of North Carolina at Chapel Hill**  
 Chapel Hill, North Carolina

**The University of North Carolina at Charlotte**  
 Charlotte, North Carolina

**The University of North Carolina at Greensboro**  
 Greensboro, North Carolina

**The University of North Carolina at Wilmington**  
 Wilmington, North Carolina

**The University of North Carolina at Pembroke**  
 Pembroke, North Carolina

**The University of North Carolina at Asheville**






- If users would like to purchase games, music downloads, books, etc., they must be able to do so using credit/debit cards, PayPal accounts. Users should also be able to barter or exchange things among themselves (e.g., exchange a \$25 worth iTunes e-coupon for \$25 Starbucks card, or gift a friend e-tickets to a movie). There probably has to be some minimal validation of the transaction (e.g. check if credit card number is 16-digit, not alphanumeric, etc.): (30 points)

[Back to Account Page](#) | Add a new card below:

Cardholder's First Name:

Cardholder's Last Name:

Credit Card Number:

Expiration Date:

CSC:

What's a CSC?

Billing Address:

Billing Address 2:

City/Town:

State/Province/Region:

Zip/Postal Code:

Country:

- User should be able to set and control their privacy settings: block other users and selectively decide what they reveal about their profiles, activities, etc. to whom: (30 points)

Sharing on Facebook

	Everyone	Friends of Friends	Friends Only
My status, photos, and posts	*		
Bio and favorite quotations	*		
Family and relationships	*		
Photos and videos I'm tagged in		*	
Religious and political views	*		
Birthday		*	
Can comment on posts			*
Places I check in to [?]			*
Contact information			*
<input checked="" type="checkbox"/> Let friends of people tagged in my posts see them.			*

[Customize settings](#) ✔ This is your current setting.

### Choose Your Privacy Settings ▶ Block Lists

[← Back to Privacy](#)

**Block users** Once you block someone, that person can no longer be your friend on Facebook or interact with you (except within applications and games you both use).

Name:  [Block This User](#)

Email:  [Block This User](#)

You haven't added anyone to your block list.

---

**Block application invites** Once you block application invites from someone, you'll automatically ignore future application requests from that friend. To block invites from a specific friend, click the "Ignore All Invites From This Friend" link under your latest request.

Block invites from:

You haven't blocked invites from anyone.

---

**Block event invites** Once you block event invites from someone, you'll automatically ignore future event requests from that friend.

Block invites from:

6. Users should be able to create events and invite people selectively: (30 points)

#### Create an Event

When?

End Time:

What are you planning?

Where?   
[Add street address](#)

More info?

Who's invited? [Select Guests](#)

Anyone can view and RSVP (public event)

Show the guest list on the event page

[Create Event](#)

7. In this section, you are required to issue SQL statements to your database and attach the result with the project you turn in. The SQL statements you need to issue are as follows: (50 points)

- Insert statements to populate all of the tables you created.
- List all friends and contact info. of a particular user, Bob
- List all events posted by user Alice between 10/1/10 to 10/30/10 and the number of people who accepted Alice's invitation to attend each event.
- List the number and names of items (gifts, e-coupons) exchanged/purchased by user Bob from user Alice.

- List the names and e-mail addresses of people whose friend requests and event invites Alice has blocked.

This project is divided into steps. The following are the steps along with deliverables you need to turn-in for each step.

**Step 1 (40 points):** Use your brainstorming skills to guess what pieces of information the system requires and you would need to store in the database. The deliverable of this step would include the list of all the pieces of information that you think the database should store and justifications for each.

**Step 2 (40 points):** In this step you would actually delve into the database design process. Organize the information you got from Step 1 into entities, attributes and the relationships among the entities. The deliverable for this step would be the complete E-R diagram depicting your E-R model. You can use any ER/UML based design tools to generate the E-R diagram.

**Step 3 (40 points):** In this phase you would actually implement your design in the database, i.e. you would be creating the database objects like tables and constraints etc. in the database management system you are given access to. The deliverable of this step would be a list of all the tables in the database and the create table statements you used. If you are using any constraints in your design, make sure you state that in the document you submit. Submit the list of tables, statements, and query results.

**Extra Credit (Optional):** If you design web interfaces, you can get up to 60 points total extra credit (10 points for each interface).

**Project Demo and Report:** You will need to do a working demonstration of the project to the TA during pre-finals week (Dec. 6 – Dec. 10). There will be 1-2 days assigned for demos, which will be decided by the class (whichever days are most convenient to all students). Each demo will last 20 minutes. There will be a sign-up sheet provided with a set of time-slots in which you would need to book or sign-up for the specific time you want. In addition, you will have to submit a project report which includes - ER design, relational database design, query answers, and user interfaces. During the demo, and in the project report, each person's roles and the work done by each member of the group have to be clearly indicated.

Submit all the required documents during pre-finals week before the demo through the "project" command. The class is 5350s001 and the project name is "SNDB".