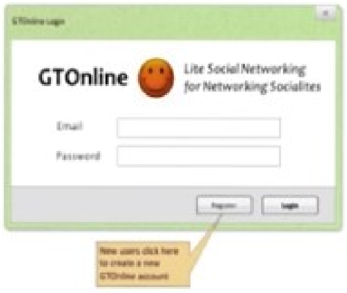


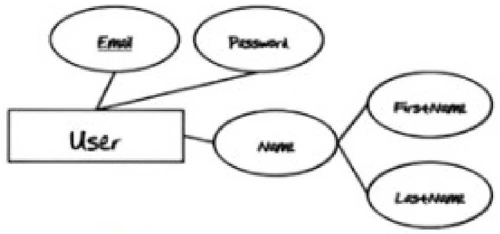
GTOnline Login

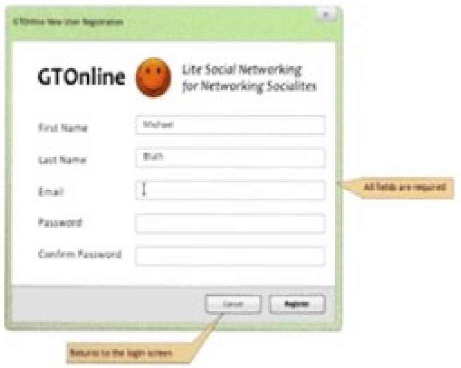
GTOnline — Lite Social Networking for Networking Socialites

Email:

Password:

Requirement: ...All Users are uniquely identified by his or her Email Address. Providing a valid Email Address and Password combination will log the user into the system...



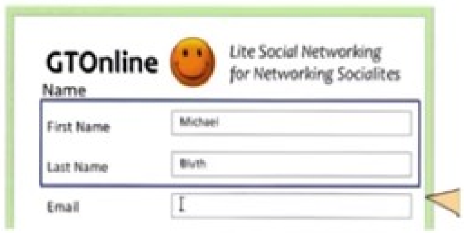


GTOnline New User Registration

GTOnline — Lite Social Networking for Networking Socialites

First Name	Michael
Last Name	Bluth
Email	
Password	
Confirm Password	

Question: Should Name be a composite attribute? Well, you decide, sort of...



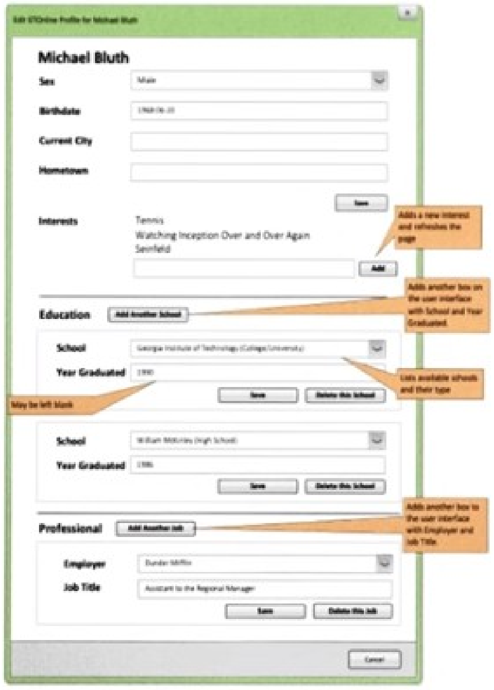
GTOnline — Lite Social Networking for Networking Socialites

Name

First Name	Michael
Last Name	Bluth
Email	

These snippets of task requirements are taken from a comprehensive project description and are just part of what is used to design the ER Diagram.

上圖中上面的是 log in screen, 下面的是 registration screen.



Edit GTOnline Profile for Michael Bluth

Michael Bluth

Sex	Male
Birthdate	1968-06-20
Current City	
Hometown	
Interests	Tennis Watching Inception Over and Over Again Seinfeld

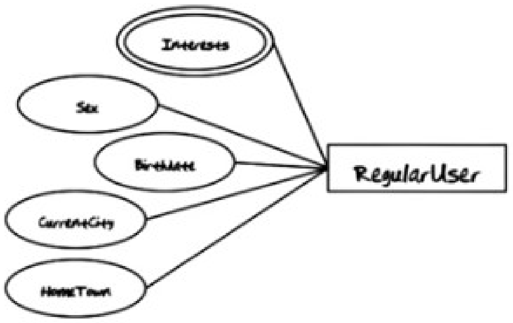
Education

School	Georgia Institute of Technology (College/University)
Year Graduated	1990
School	William McKinley (High School)
Year Graduated	1986

Professional

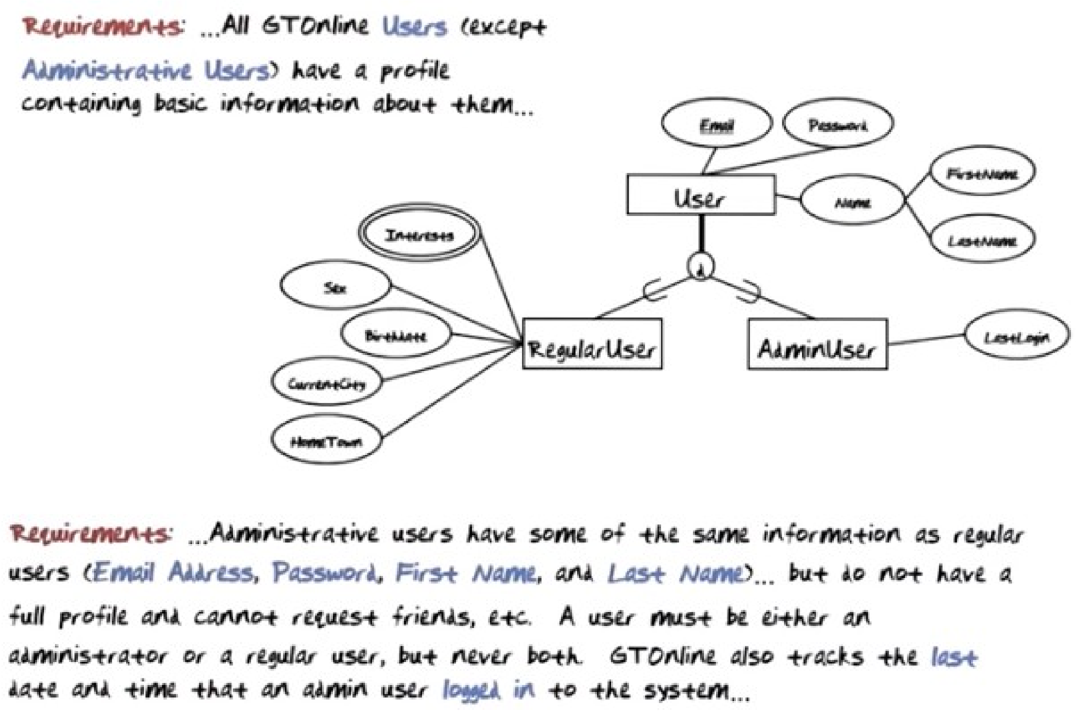
Employer	Dunder Mifflin
Job Title	Assistant to the Regional Manager

Save
Add
Adds a new interest and refreshes the page
Add Another School
Adds another box on the user interface with School and Year Graduated
Lists available schools and their type
May be left blank
Save
Delete this School
Add Another Job
Adds another box to the user interface with Employer and Job Title
Delete this Job
Cancel


Interests
Sex
Birthdate
CurrentCity
HomeTown
RegularUser

Use attribute names from the documents in the EER Diagram!

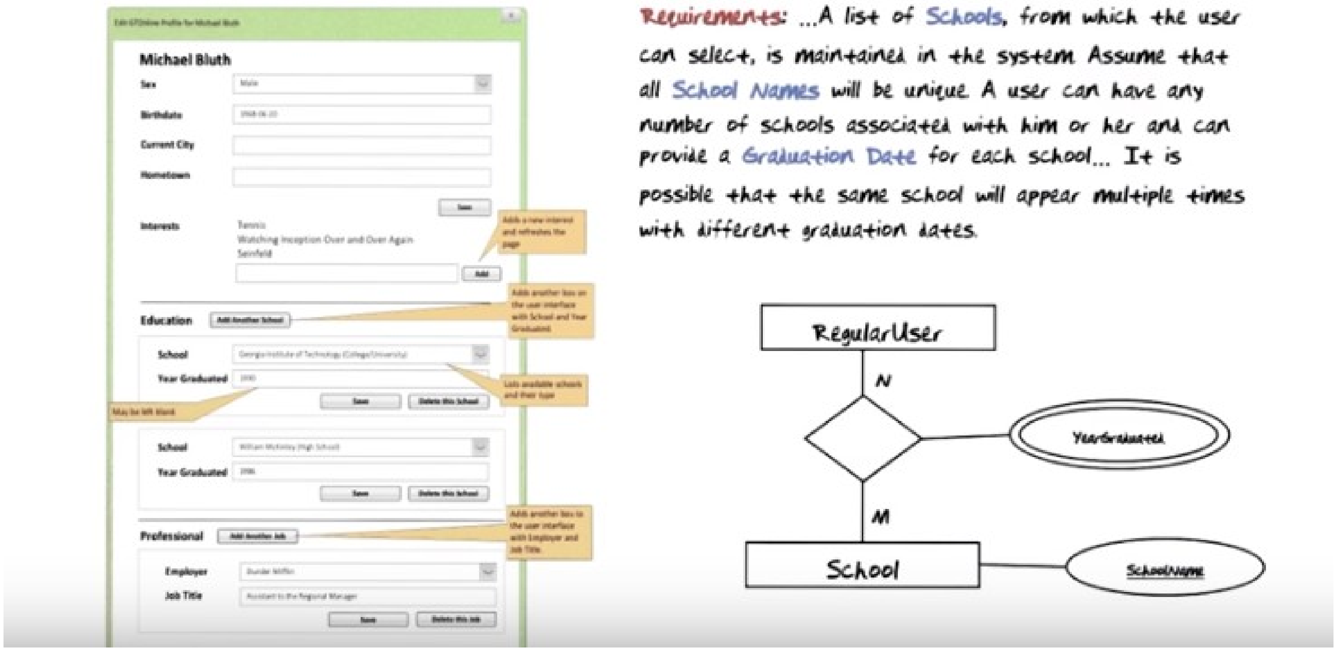
上圖是 Edit Profile screen.



Requirements: ...All GTOnline Users (except Administrative Users) have a profile containing basic information about them...

Requirements: ...Administrative users have some of the same information as regular users (Email Address, Password, First Name, and Last Name)... but do not have a full profile and cannot request friends, etc. A user must be either an administrator or a regular user, but never both. GTOnline also tracks the last date and time that an admin user logged in to the system...

The above figure: a user must either be a Regularuser or an AdminUser, but cannot be both. Therefore, the constraint is disjoint. Textbook 中說了, d 下面兩個箭頭(或 U)表示 inherit 的方向(此處是除 quiz 外, 最早出現箭頭的地方)



Edit GTOnline Profile for Michael Bluth

Michael Bluth

Sex	Male
Birthdate	1968-06-22
Current City	
Hometown	

Save

Interests: Tennis; Watching Inception Over and Over Again; Seinfeld

Add

Education — Add Another School

School	Georgia Institute of Technology (College/University)
Year Graduated	1990

Save — Delete this School

School	William McKinley High School
Year Graduated	1986

Save — Delete this School

Professional — Add Another Job

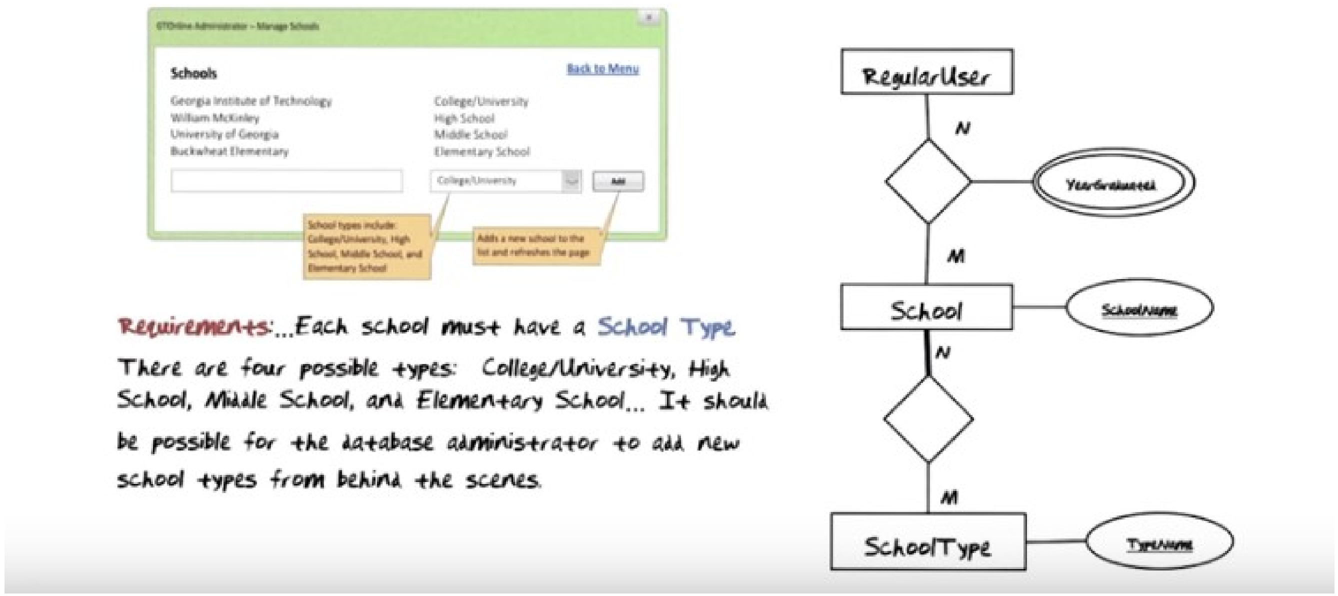
Employer	Dunder Mifflin
Job Title	Assistant to the Regional Manager

Save — Delete this Job

Adds a new interest and refreshes the page
Adds another box on the user interface with School and Year Graduated
Lists available schools and their type
May be left blank
Adds another box to the user interface with Employer and Job Title

Requirements: ...A list of Schools, from which the user can select, is maintained in the system Assume that all School Names will be unique. A user can have any number of schools associated with him or her and can provide a Graduation Date for each school... It is possible that the same school will appear multiple times with different graduation dates.

上圖: Edit Profile 往下看, 就是 Education section.



Schools

Georgia Institute of Technology	College/University
William McKinley	High School
University of Georgia	Middle School
Buckwheat Elementary	Elementary School

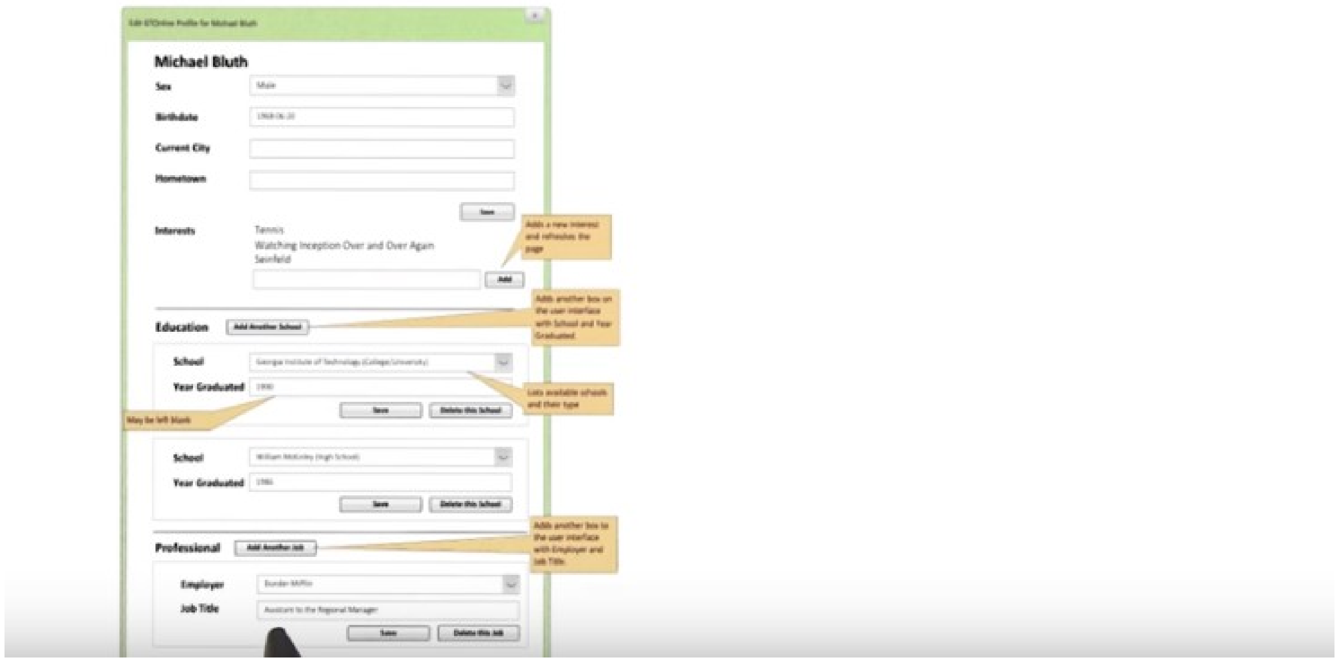
College/University

School types include: College/University, High School, Middle School, and Elementary School

Adds a new school to the list and refreshes the page

Requirements:...Each school must have a School Type. There are four possible types: College/University, High School, Middle School, and Elementary School... It should be possible for the database administrator to add new school types from behind the scenes.

上圖: relation 連雙線 property 的, 都可以這樣理解: 一個 RegularUser 和一個 School 可以對應多個 YearGraduate, 即同一個人在同一個學校中可以有多個不同的畢業時間. 下下圖中的 RegularUser-Employer-JobTitle 也可以這樣理解. relation 連單線 property 的, 也可以類似理解.



Michael Bluth

Sex	Male
Birthdate	1968-06-20
Current City	
Hometown	
Interests	Tennis Watching Inception Over and Over Again Seinfeld

Adds a new interest and refreshes the page

Education

Adds another box on the user interface with School and Year Graduated

School	Georgia Institute of Technology (College/University)
Year Graduated	1990

Lists available schools and their type

May be left blank

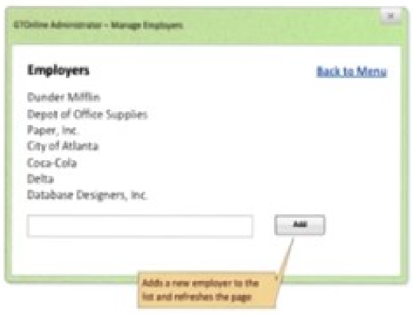
School	William McKinley (High School)
Year Graduated	1986

Professional

Adds another box to the user interface with Employer and Job Title

Employer	Dunder Mifflin
Job Title	Assistant to the Regional Manager

The above figure: at the bottom of the Edit Profile screen is the Professional section.



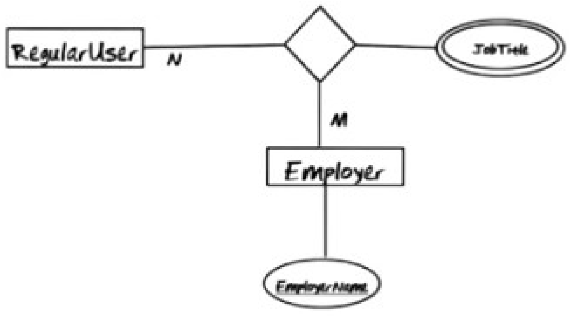
GTOnline Administrator – Manage Employers

Employers Back to Menu

Dunder Mifflin
Depot of Office Supplies
Paper, Inc.
City of Atlanta
Coca-Cola
Delta
Database Designers, Inc.

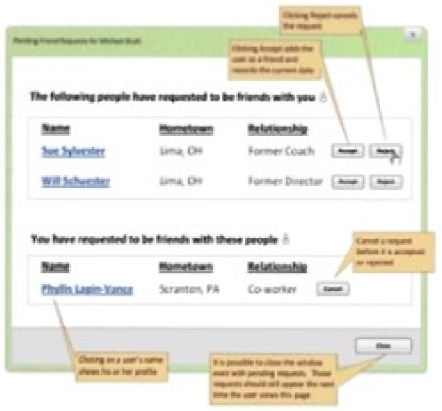
Add

Adds a new employer to the list and refreshes the page



Requirements: ...Administrators are responsible for managing the list of Employers. Assume that all Employers have a unique Name

Requirements: ...The Job Title field is not managed by the administrator and can be any value provided by the user. A profile can contain multiple Employers and the same Employer may even appear multiple times as long as the Job Title is different in each case



Pending Friend Requests for Michael Bluth

The following people have requested to be friends with you

Name	Hometown	Relationship		
Sue Sylvester	Lima, OH	Former Coach	Accept	Reject
Will Schuester	Lima, OH	Former Director	Accept	Reject

You have requested to be friends with these people

Name	Hometown	Relationship	
Phyllis Lapin-Vance	Scranton, PA	Co-worker	Cancel

Close

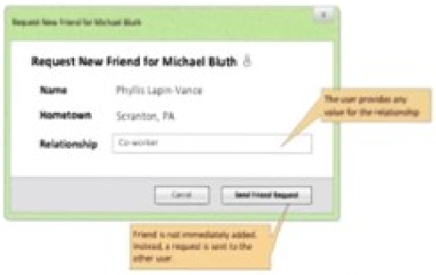
Clicking Reject cancels the request

Clicking Accept adds the user as a friend and records the current date

Cancel a request before it is accepted or rejected

Clicking on a user's name shows his or her profile

It is possible to close the window even with pending requests. Those requests should still appear the next time the user views this page.



Request New Friend for Michael Bluth

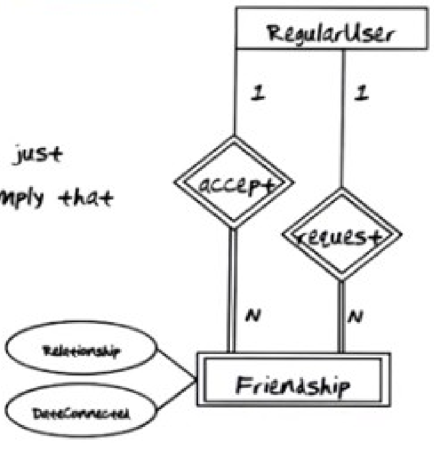
Request New Friend for Michael Bluth

Name	Phyllis Lapin-Vance
Hometown	Scranton, PA
Relationship	Co-worker

Cancel Send Friend Request

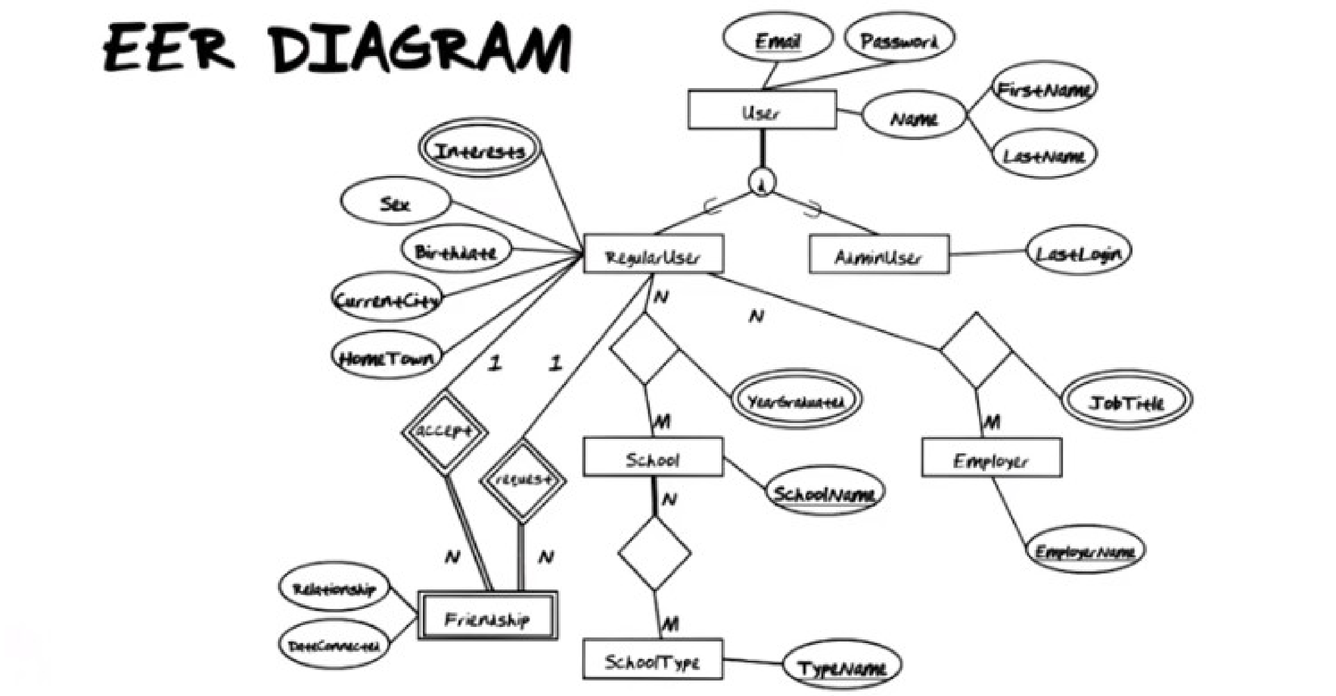
The user provides any value for the relationship

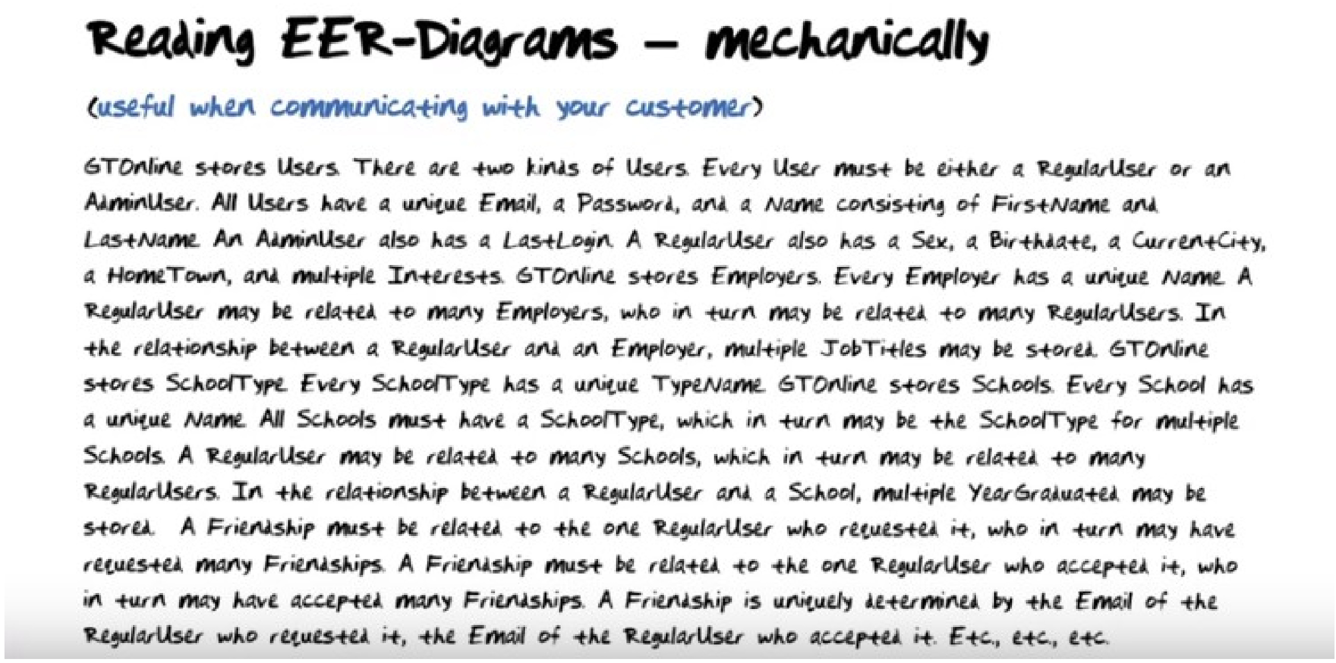
Friend is not immediately added. Instead, a request is sent to the other user



Requirements: ...Friendship is not always reciprocal... just because Emily is friends with Sarah, this does not imply that Sarah is friends with Emily...

...the DateConnected field is set when the friend request is accepted, not when the request is originally sent.



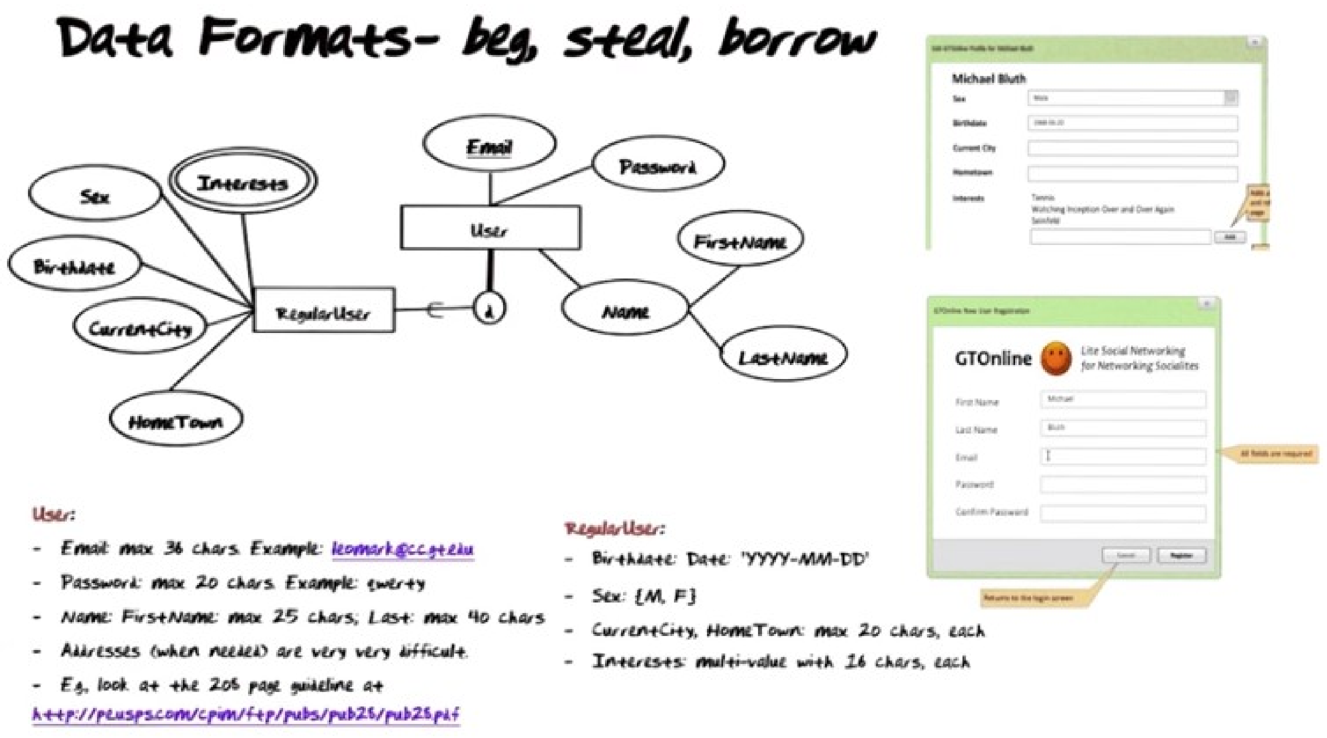


Reading EER-Diagrams – mechanically

(useful when communicating with your customer)

GTOnline stores Users. There are two kinds of Users. Every User must be either a RegularUser or an AdminUser. All Users have a unique Email, a Password, and a Name consisting of FirstName and LastName. An AdminUser also has a LastLogin. A RegularUser also has a Sex, a Birthdate, a CurrentCity, a HomeTown, and multiple Interests. GTOnline stores Employers. Every Employer has a unique Name. A RegularUser may be related to many Employers, who in turn may be related to many RegularUsers. In the relationship between a RegularUser and an Employer, multiple JobTitles may be stored. GTOnline stores SchoolType. Every SchoolType has a unique TypeName. GTOnline stores Schools. Every School has a unique Name. All Schools must have a SchoolType, which in turn may be the SchoolType for multiple Schools. A RegularUser may be related to many Schools, which in turn may be related to many RegularUsers. In the relationship between a RegularUser and a School, multiple YearGraduated may be stored. A Friendship must be related to the one RegularUser who requested it, who in turn may have requested many Friendships. A Friendship must be related to the one RegularUser who accepted it, who in turn may have accepted many Friendships. A Friendship is uniquely determined by the Email of the RegularUser who requested it, the Email of the RegularUser who accepted it. Etc., etc., etc.

The above figure: Please notice that the way I just summarized Extended-Entity Relationship diagram was quite mechanical in nature. It would actually be possible to write a program that would take as input an Extended-Entity Relationship diagram and then read out in English exactly what it means, no more no less. When you are talking with your customer about your database specificaiton, they would not be able to understand the Extended-Entity Relationship diagram, they might be able to understand this English representation of that diagram.



Michael Bluth

Sex	Male
Birthdate	1968-06-22
Current City	
Hometown	
Interests	Tennis Watching Inception Over and Over Again Seinfeld

GTOnline Lite Social Networking for Networking Socialites

First Name	Michael
Last Name	Bluth
Email	
Password	
Confirm Password	

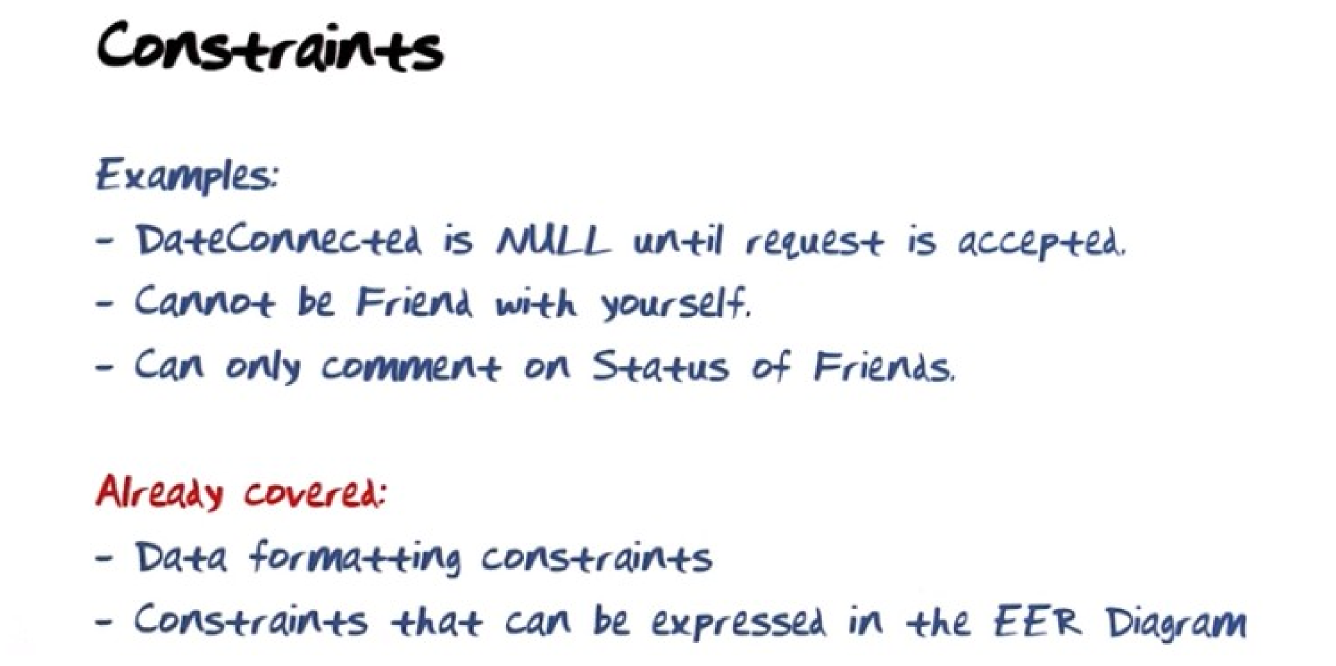
All fields are required

Returns to the login screen

User:
- Email: max 36 chars. Example: leomark@cc.gt.edu
- Password: max 20 chars. Example: qwerty
- Name: FirstName: max 25 chars; Last: max 40 chars
- Addresses (when needed) are very very difficult.
- E.g., look at the 206 page guideline at http://pe.usps.com/cpim/ftp/pubs/pub28/pub28.pdf

RegularUser:
- Birthdate: Date: 'YYYY-MM-DD'
- Sex: {M, F}
- CurrentCity, HomeTown: max 20 chars, each
- Interests: multi-value with 16 chars, each

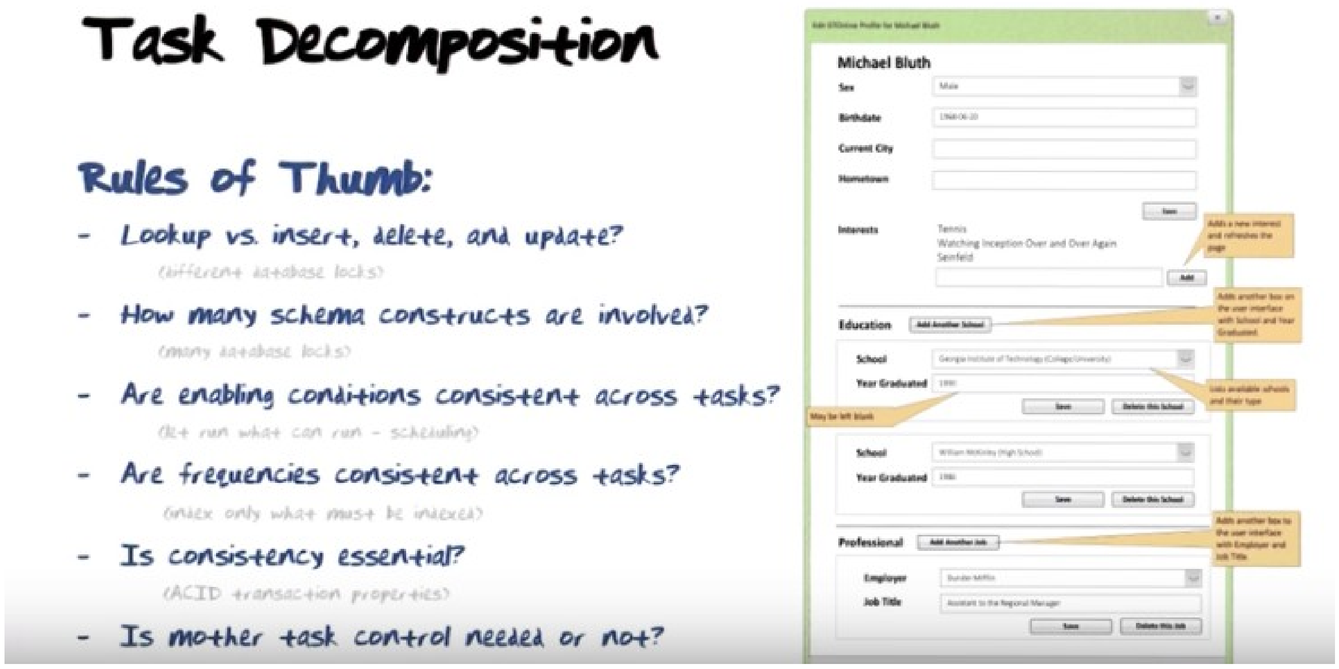
The above figure: Addresses, when needed, are very very difficult to deal with. Couple of years agao when I was looking at address formats, I ran into the United States Postal Service's recommendation for addresses across the world. It's a whopping two hundred-plus page guideline. Guidelines like that takes specilaists to develop, and it takes a lot of time. Instead of every single database to sing out there who needs to create addresses in their database, maybe they should take a look at what other people have done. In general, you can extrapolate their idea into one of the basic concept when it comes to data formatting, namely beg, steal and borrow. If someone already has spent a lot of time thinking about this, why would you do it too?


Constraints

Examples:
- DateConnected is NULL until request is accepted.
- Cannot be Friend with yourself.
- Can only comment on Status of Friends.

Already covered:
- Data formatting constraints
- Constraints that can be expressed in the EER Diagram

The above figure: Well, it basically leaves us with a set of rules that we can't describe in the database that we can't format out way out of. So there's only one place left to it, and that is to program it into the application program, show that constraints like this are enforced.


Task Decomposition

Rules of Thumb:
- Lookup vs. insert, delete, and update?
  (different database locks)
- How many schema constructs are involved?
  (many database locks)
- Are enabling conditions consistent across tasks?
  (let run what can run - scheduling)
- Are frequencies consistent across tasks?
  (index only what must be indexed)
- Is consistency essential?
  (ACID transaction properties)
- Is mother task control needed or not?

Michael Bluth

Sex	Male
Birthdate	[illegible]
Current City	
Hometown	
Interests	Tennis Watching Inception Over and Over Again Seinfeld

Education

School	Georgia Institute of Technology (College/University)
Year Graduated	[illegible]
School	William McKinley (High School)
Year Graduated	[illegible]

Professional

Employer	Dunder Mifflin
Job Title	Assistant to the Regional Manager

Adds a new interest and refreshes the page

Adds another box on the user interface with School and Year Graduated

Lists available schools and their type

May be left blank

Adds another box to the user interface with Employer and Job Title

The above figure: for each one of the tasks, we identify in the information flow diagram, we need to talk about whether that's a single task, or whether it should be decomposed. The rules of thumb I'm going to give you are based on a deep understanding of the internal workings of a database management system. In each one of the cases I'm going to give you the rule, and I'm going to tell you what it is based on. Later on, when you do task decompostion, you're going to go through these rules that I'm giving you in order to determine whether the particular task you're looking at should be decomposed or not.

- Lookup vs insert, delete and update?

So from a database management system perspective, difference between whether a task just is a look up task or whether it actually modifies the database through an insertion, deletion, or update. The reason for that is that different database locks are needed in order to support lookups versus chan es to the database. So if many different things take place(即需要 look up), then that's a indication that the test should be decomposed.

- How many schema constructs are involved?

The second rule of thumb is based on how big a portion of the database is involved in the operation. The bigger the portion is, the harder it is to acquire all the logs that are needed in order to support concurrent execution of transactions on the database. So if a large number of schema constructs are involved, that would make you want to decompose the task into smaller tasks using smaller portions of the database. (Edit Profile 例子中, Clearly there are several different schema constructs involved. There's both the personal information portion for user and for regular user. There's the whole education portion. And there's the whole professional portion. 多個 schema 原來就是這個意思)

- Are enabling conditions consistent across tasks?

The next rule of thumb is based on what enables the different portions of a task. So if some portion of a task are enabled, then why not let those proceed instead of waiting to later on when there might not be resource available to run those tasks that were earlier enabled. The smaller portions of a task that can run, the easier it is to schedule the task. (View profile 例子中, All three are enabled by a user's login or a friend's lookup)

- Are frequencies consistent across tasks?

The next rule of thumb is based on what the frequencies of the things that are done by the task are. If the task contains things that are done with a high frequency and things that are done with a low frequency, it is a good idea to split them apart. The reason is that high frequency things, you want to index to run fast. Whereas low frequency things, you may not need to index.

- Is consistency enssential?

The next really important rule of thumb is related to consistency of the database. Is it really essential that all the pieces of a task get done in one transaction? Or is it okay that things get spread out a little bit over time? An example of that would be transferring money from one bank account to another one. (轉出和轉入要同時做, 因為不能讓轉出的帳戶和轉入的帳戶都同時有同一托錢). So it is very important that all the steps involved are done together. For example, monies must be available in the from account. They must be subtracted from the balance of the from account, and they must be deposited into the to account. All of those things must be done together, in order that you do not end up with an inconsistent database.

- Is mother task control needed or not?

The final rule of thumb is based on the question of whether a mother task is needed to control subtasks. Sometimes you need a mother task, sometimes you do not. The banking example I just gave you certainly needs a mother task (View profile 例子中, all three of the must be done in order to display the information on the View Profile screen. So a mother task is needed to make sure that all three things get done). We're later going to see things that may not need a mother task.

Web-apps

- Traditionally, almost stateless.
- Must have some state, eg email of the session user.
- May need some click stream history.
- Things are changing. So-called Web 2.0 and AJAX technologies provide more rich user interface in the Web browser. These technologies make it easier to keep state information locally on the browser or behind the scenes on the server (but not in the DB).
- In this sense, the web apps are beginning to act more like traditional apps.

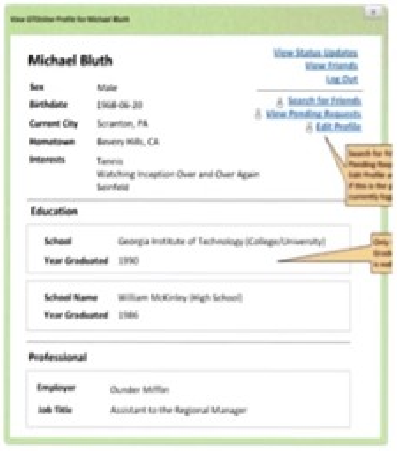
Traditional-apps

- In a traditional app, it is much easier to manage local state separately from the DB (eg, using smart widgets, etc.).
- A whole slew of changes can be collected before submitting them all to the database.
- Supports better control of ACID transactions execution.

Task Decomposition ViewProfile

View Profile:

- three lookups of Personal, Education, and Professional information for a RegularUser
- All three are read-only.
- All three are enabled by a user's login or a friend's lookup.
- All three have the same frequency.
- Several different schema constructs are needed.
- Consistency is not critical, even if the profile is being edited by the user while a friend is looking at it.
- They can be done in any order.
- All three must be done, so mother task is needed.
- <u>Should be decomposed into three sub-tasks.</u>


View GTOnline Profile for Michael Bluth

Michael Bluth

View Status Updates
View Friends
Log Out

Search for Friends
View Pending Requests
Edit Profile

Sex	Male
Birthdate	1968-06-20
Current City	Scranton, PA
Hometown	Bevery Hills, CA
Interests	Tennis Watching Inception Over and Over Again Seinfeld

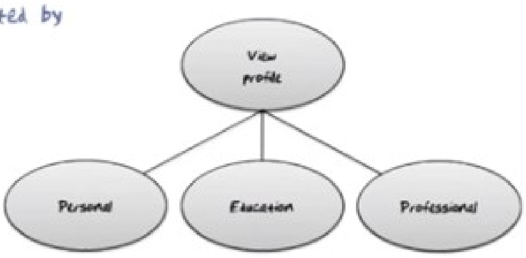
Education

School	Georgia Institute of Technology (College/University)
Year Graduated	1990

School Name	William McKinley (High School)
Year Graduated	1986

Professional

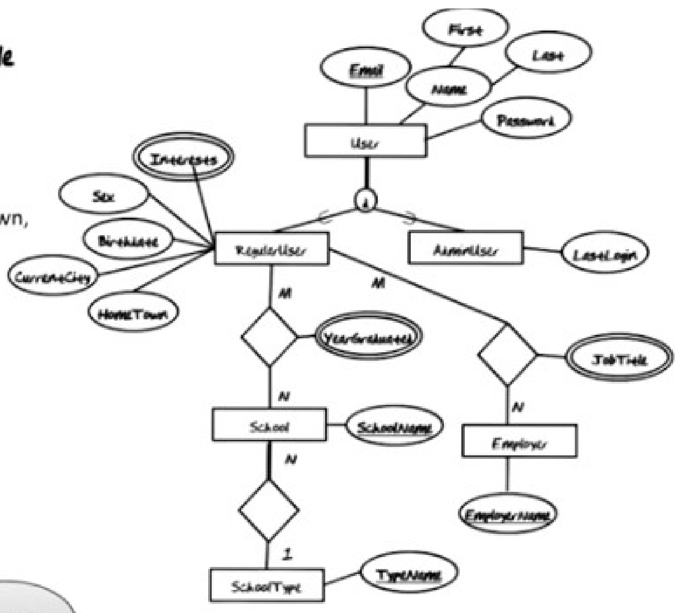
Employer	Dunder Mifflin
Job Title	Assistant to the Regional Manager

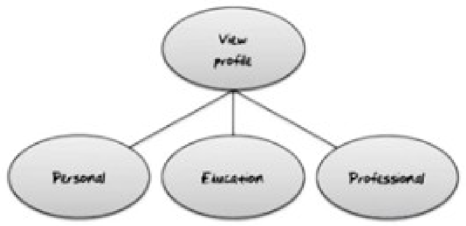


Abstract Code ViewProfile

View Profile

```
Find the current User using the User Email;
Display User Name;
Find the current RegularUser using the User Email;
Display RegularUser Sex, Birthdate, CurrentCity, Hometown,
and Interests;
Find each School for the RegularUser
    {Display School Name and YearsGraduated;
    Find SchoolType;
    Display SchoolType Name};
Find each Employer for the RegularUser
    Display Employer Name and JobTitles;
```

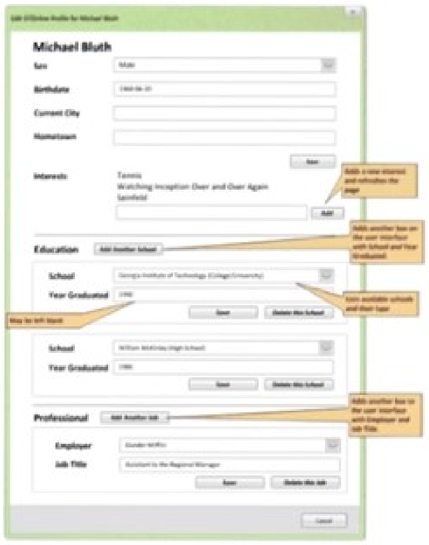




Task Decomposition EditProfile

Edit Profile:

- Lookups of Personal, Education, and Professional information of a RegularUser (use: **View Profile**)
- Lookups of School and Employer lists
- Edits of Personal, Education, and Professional information
- Read, insert, delete, and update
- All three are enabled by a user's login and separate edit request.
- Different frequencies
- Several different schema constructs are needed.
- Consistency is not critical, even if the profile is being looked at by a friend of the user
- Lookup done first followed by any number of edits and lookups
- Mother task is needed.
- ***Must be decomposed into sub-tasks***



Michael Bluth

Sex	Male
Birthdate	[illegible]
Current City	
Hometown	
Interests	Tennis Watching Inception Over and Over Again Seinfeld

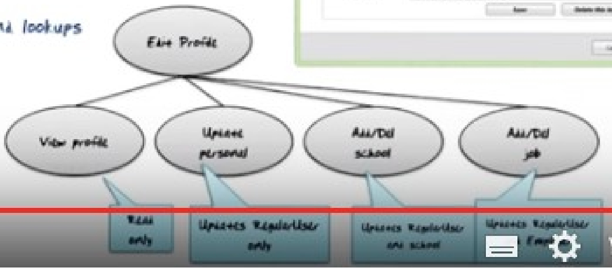
Education

School	Georgia Institute of Technology (College/University)
Year Graduated	[illegible]

School	[illegible]
Year Graduated	[illegible]

Professional

Employer	[illegible]
Job Title	[illegible]



3:40 / 3:40

Abstract Code

Edit Profile

<u>Edit Profile</u>

View Profile. Populate School and Employer dropdowns.

While no buttons are pushed, do nothing.

When a button is pushed, then do the following:

{If *SAVE PERSONAL:* **Update Personal**. **View Profile.**

If *DELETE THIS SCHOOL*: **Delete School**. **View Profile.**

If *DELETE THIS JOB*: **Delete Job**. **View Profile.**

If *ADD ANOTHER SCHOOL*: **View Profile;** display the form with room for another School;

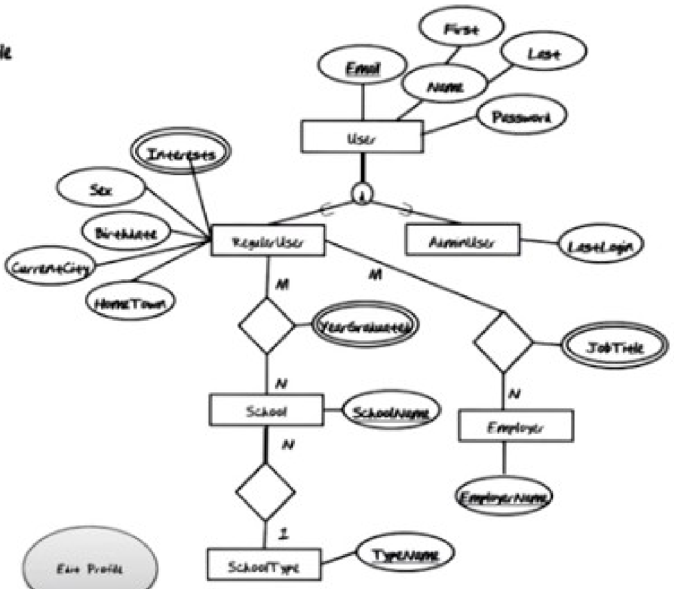
If *ADD ANOTHER JOB*: **View Profile;** display the form with room for another Job

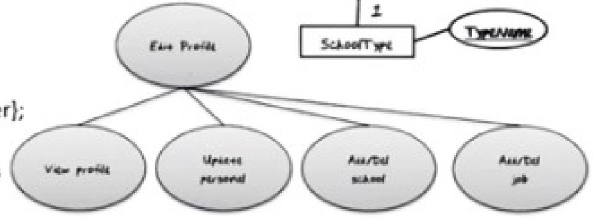
If *SAVE SCHOOL*: **Add School**. **View Profile**

If *SAVE JOB*: **Add Job**. **View Profile**

If *CANCEL*: Go to **View Profile** for current User};

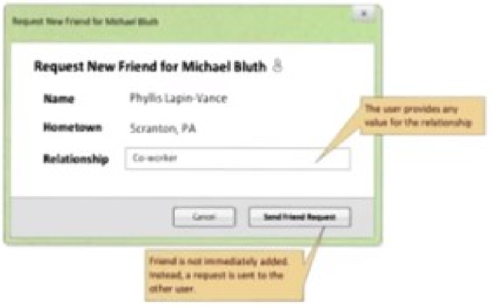
BoldUnderline: Task definition. **Bold**: Task call. *Italics*: Button names





Task Decomposition

Friend Requests etc.



Request New Friend for Michael Bluth

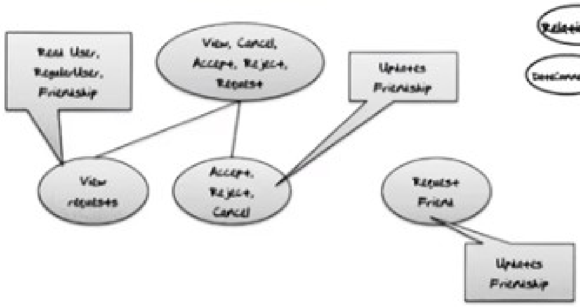
Request New Friend for Michael Bluth

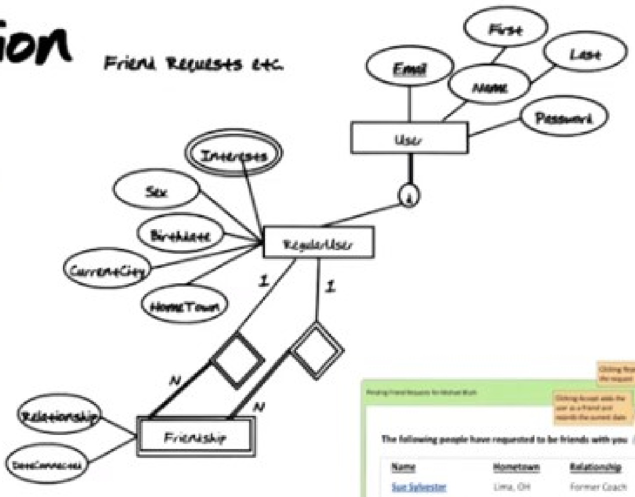
Name	Phyllis Lapin-Vance
Hometown	Scranton, PA
Relationship	Co-worker

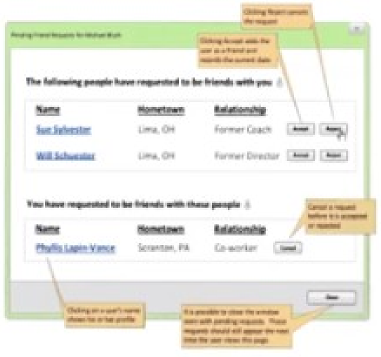
The user provides any value for the relationship

Cancel | Send Friend Request

Friend is not immediately added. Instead, a request is sent to the other user.







Pending Friend Requests for Michael Bluth

Clicking Reject cancels the request

Clicking Accept adds the user as a friend and records the current date

The following people have requested to be friends with you

<u>Name</u>	<u>Hometown</u>	<u>Relationship</u>		
Sue Sylvester	Lima, OH	Former Coach	Accept	Reject
Will Schuester	Lima, OH	Former Director	Accept	Reject

You have requested to be friends with these people

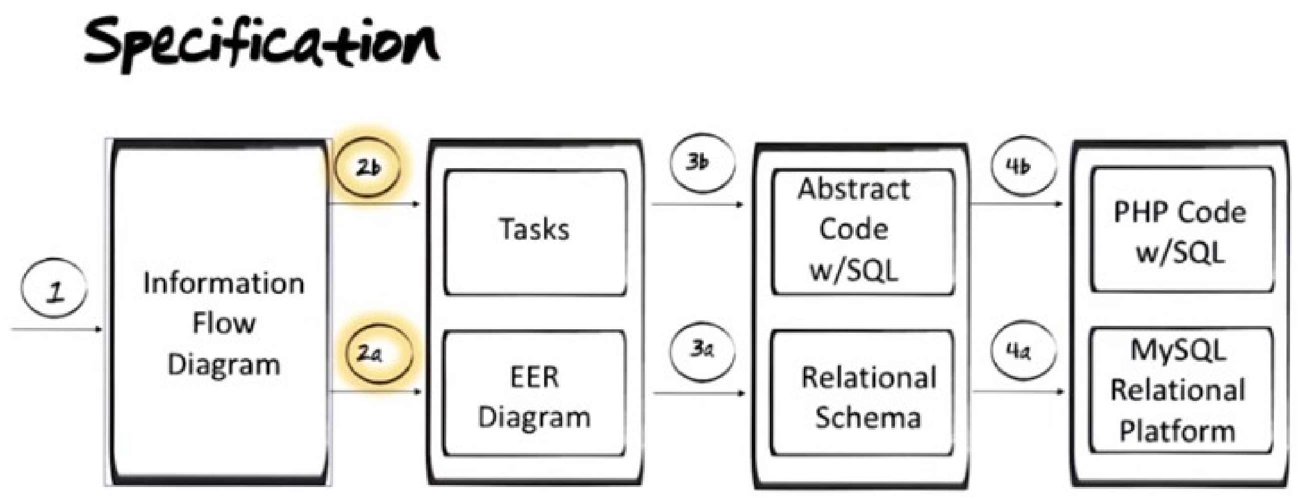
Cancel a request before it is accepted or rejected

<u>Name</u>	<u>Hometown</u>	<u>Relationship</u>	
Phyllis Lapin-Vance	Scranton, PA	Co-worker	Cancel

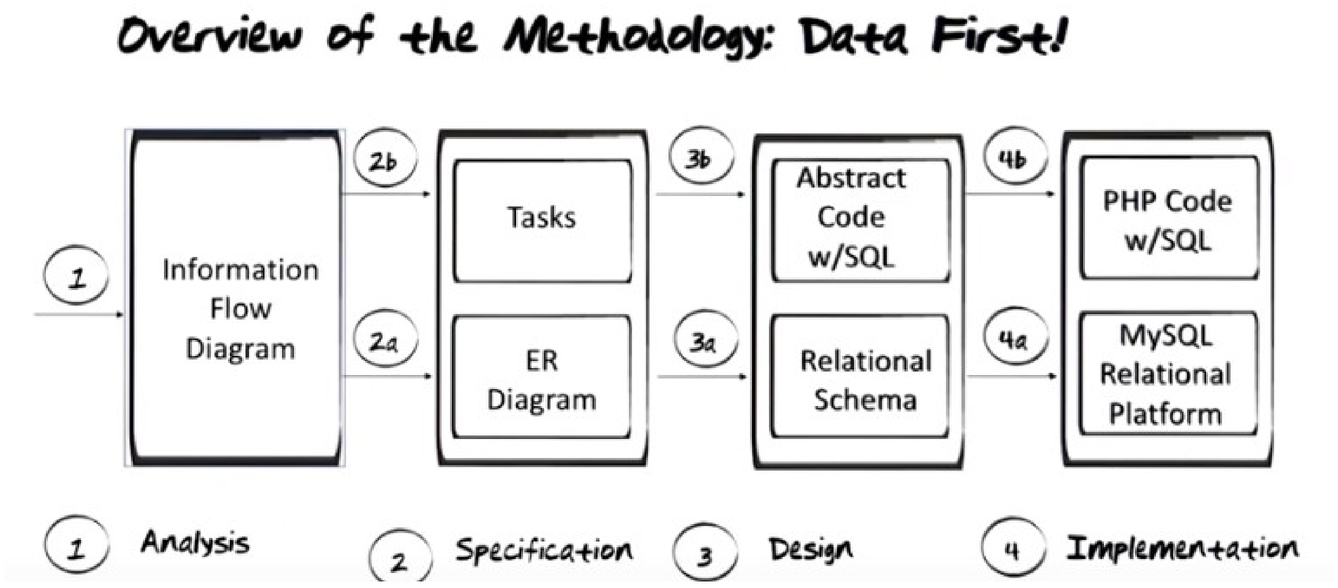
Close

Clicking on a user's name shows his or her profile

It is possible to close the window even with pending requests. These requests should still appear the next time the user views this page.



其實看之前的原圖更清楚:



The figure above: Before we go any further, we need a small detour to study an additional database tool or technique.