LawsForMe

Steven Landau, Tory Leo, Talha Azhar

September 29, 2017

* Overview

LawsForMe™ is a web application that will inform users (via email) of laws that exist and that are being passed/proposed that might aﬀect their interests. When someone registers an account on the LawsForMe website their address will be used to determine their state and county. The user is assumed to be in the United States. The user will then enter their interests which will result in Federal, State, and County laws being scanned for possible laws that aﬀect their interest (positively or negatively). There will also be a forum section of the website where users will be able to discuss the impacts of various laws on various hobbies.

1.5 Phase 3 Overview

Significant progress was made during phase 3. The website changed massively in terms of aesthetic, feel, and design. A new template was introduced to make the website look more professional and help us with the tone we are trying to establish. The website is now much smoother as both the database and queries have been optimized. Many new queries have been added to the program. Many of the new queries relate to the forum portion of the website but there are many that serve to obtain the interests of the users as well as the laws that affect said interests. Determining what laws affect a user’s interest happen asynchronously when a sign up. New tables have also been created to relate interests to laws that affect them.

For every interest, there are multiple laws that affect it. For every law, there are multiple interests that the law affects. A one to many relationships was established between laws and interests due to this relationship. Users are capable of having multiple interests so a one to many relationships was established between Users and their interests. Essentially the table that was created that keeps tracks of the interests of every user has “interests” as a foreign key which references a table of interests as to not repeat any data. The “user” key of that table is also a foreign key that references the user.

The forums were a major part of the project that was completed during this phase. Users are now able to browse, sort and enter discussions which contain various threads. These threads have a certain number of comments (displayed to the user) and starting\_text indicating what the thread is about. A user is able to enter (or sort) the thread and view comments that other users posted as well as post their own.

* Requirements
  + Users must be able to register, giving a minimum of their email, and state.
  + Federal and Statelaws need to parsed in search of data relating to users interersts.
  + The website must have a fully functional forum section where users can discuss various laws.
  + Users must be emailed whenever a law is being proposed that could aﬀect their interests.

4 Progression

Phase 1

* + GitHub created for LawsForMe project
  + Database created on Reddwarf with tables
  + All team members have logged into Reddwarf
  + Created UML to lay out the design of the program
  + CSS made for the project
  + HTML written for the main page. Starting writing Javascript for login
  + HTML written for the account creation. Started writing Javascript for account creation

Phase 2

* + Backend switched to Django
  + Basic database queries written
  + Tables finalized
  + Wikipedia parsed for laws
  + Account creation/registration mostly done
  + Website became usable

Phase 3

* + Account creation perfected
  + Login perfected
  + Able to determine what laws affect what interests
  + Laws that affect interests displayed on home page
  + Forums finished
  + Users can add comments to threads
  + Users can sign out at will.
  + Webpage will adapt to user state (give them the option to sign out if signed in and vice versa)

5 Use Cases

* + Name: Login

Actors: User

Entry Conditions:

1. The User is on the Login page, which means they are not currently logged in

Flow of Events:

1. User Enters Username into Username Textbox

2. User Enters Password into Password Textbox

3. User hits “submit” button and the inputted username and password are sent to server

4. Database is queried to identify if the username exists and if it does, if the password is the inputted password

5. User is redirected to an “Interest page”

Alternate Flows:

1. If Username or Password do match up with a user in the database, an error is displayed on the screen, notifying them that either the Username or Password is invalid

2. If the User forgets to enter a Username or Password, an error is shown, notifying them that they must enter a username and password

* + Name: Account Creation

Actors: User

Entry Conditions:

1. User is on the Account Creation page and doesn’t have a pre-existing account

Flow of Events:

1. User fills in the fields that are listed as “required” and (possibly) unrequired fields
2. User hits “submit”
3. Database is queried to identify that the fields entered that are to become primary keys are unique
4. User is redirected to an “Interest Page”

Alternate Flows:

1. User lists information in a primary key field, with information that is already in the database and gets a response saying that information for that field is already taken
2. User doesn’t fill out all “required” information fields and gets a response saying that they must enter information in that field
3. User enters incorrect information into a field and gets a response telling them they entered invalid data in that field
   * Name: Interest Page
     1. Actors: User
     2. Entry Conditions:
        1. 1. User is logged in and hasn’t added any interests to their interest list
        2. Flow of Events:
        3. 1. User is presented a list of interests
           1. 2. User can select interests from the list by clicking on them, which adds them to their interest list
           2. 3. User hits “submit” and is then presented with a list of laws that might affect them
           3. Alternate Flows:
           4. 1. The User has not selected any interests and is presented with an error telling them to select some interests

2. User clicks an icon that brings them to the Account Information page

1. 3. User clicks the “Logout” and is brought to the Login page
   * Name: Law Information

Actors: User

Entry Conditions:

1. User has selected interests from the list of interests and is now displayed Laws related to those interests

Flow Events:

1. User clicks on a specific law and is displayed information related to that law
2. User can hit the “back” button to go back to the page that listed relevant laws

Alternate Flows:

1. User clicks an icon that brings them back to the Interest Page
2. User clicks an icon that brings them to the Account Information page
3. User clicks the “Logout” button and is brought to the Login page
   * User can go to an information update screen to update their address
   * Name: Information Update

Actors: User

Entry Conditions:

1. User is now on the “Information Update” page

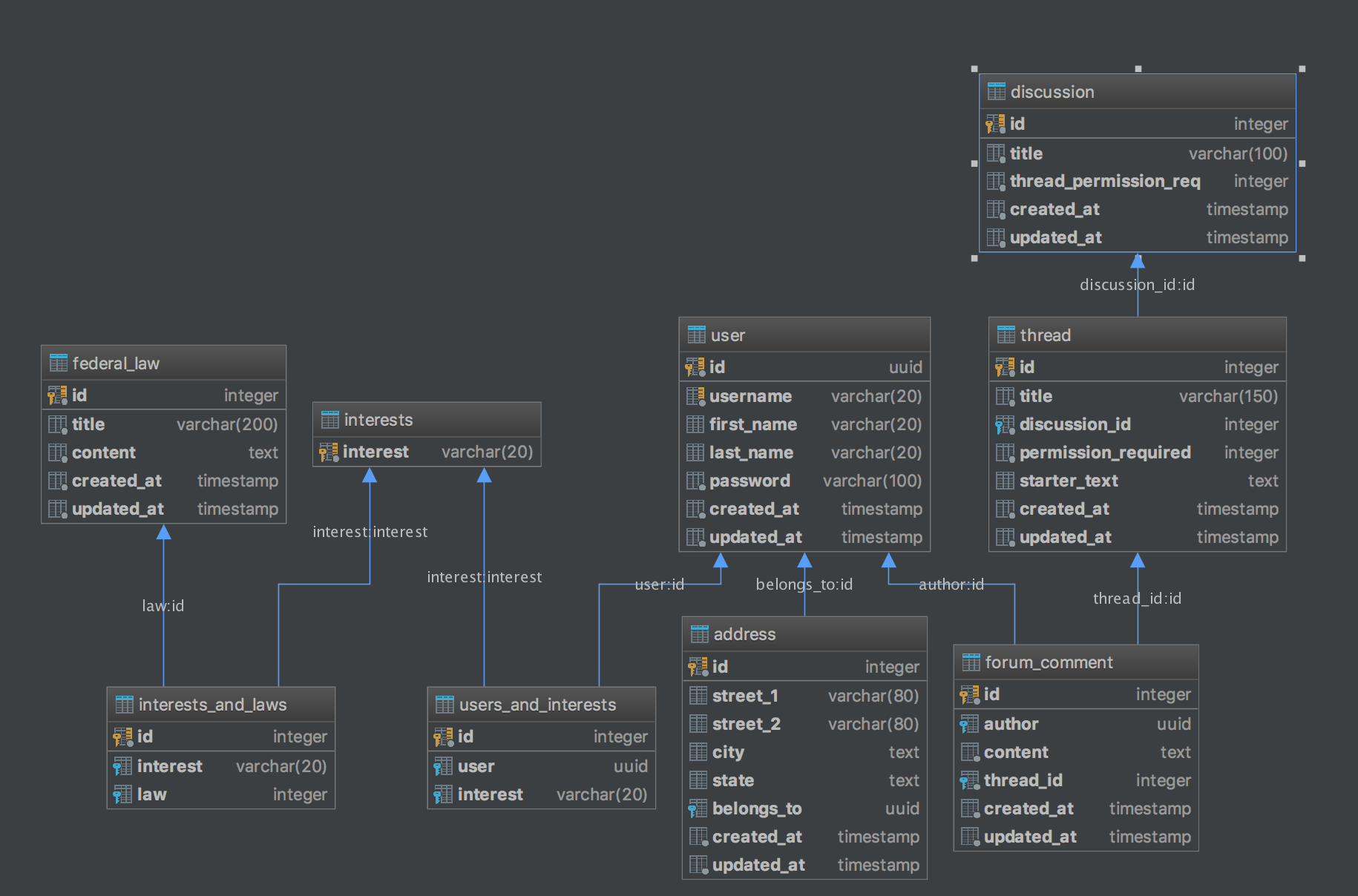
Flow Events:

1. User updates fields on their page pertaining to their information
2. User clicks “save” button and is redirected to the “Interests page”

Alternate Flows:

1. User clicks “back” button without making changes and is brought back to the “Interests” page
2. User clicks “back” button with making changes and is prompted whether if they want to save their changes and is then brought back to the “Interests” page
3. User clicks the “Logout” button and is brought to the Login page

6 Table Normalization



The database is in BCNF. In order for a database to be in BCNF the table must be in 1NF, 2NF and 3NF and must not have any duplicate data.

1NF: The table is in 1NF because all attributes are atomic. There is no way to further divide any of the attributes. Originally first\_name and last\_name were a single attribute “name”. This was not atomic so we split the attribute into two separate ones.

2NF: The table is in 2NF because it is in 1NF and there are no partial key dependencies what so ever. Each attribute in every table depends entirely on the primary key (whereas for each table there is only a single primary key).

3NF: The table is in 3NF because it is in 1NF, 2NF and there is no transitive key dependency. None of these tables have a non-primary key attribute that implies a subset of the other attributes.

BCNF: The table is in BCNF because it is in 1NF, 2NF, 3NF and has no duplicate data. There are tables such as users\_and\_interests that house both user and interest despite interest and user appearing in other tables (as a primary key). This is because both of those attributes are used as foreign keys in that table, thus the data is not repeating.

7 Table Reduction

The image in the previous section was reduced to the following tables;

User = (uuid, first\_name, last\_name, username, password, updated\_at, created\_at)

Address = (id, street\_1, street\_2, city, state, belongs\_to)

Federal\_laws = (id, content, title)

Thread = (id, title, discussion\_id, permission\_required, starter\_text)

Forum\_comment = (id, author, content, author, thread\_id)

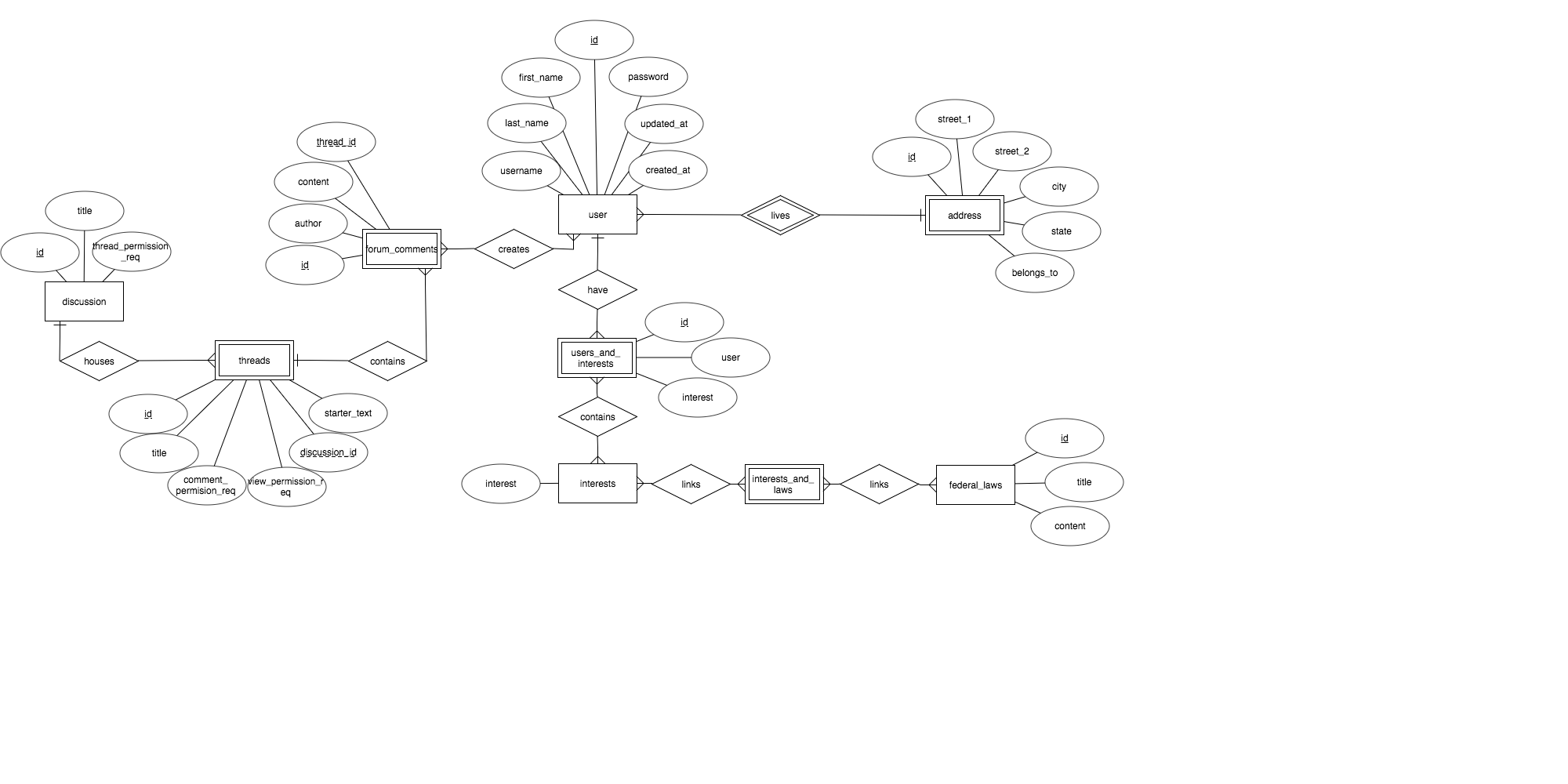
Discussion = (id, title, thread\_permission\_req)

Interests = (interest)

Users\_and\_interests = (id, user, interest)

Interests\_and\_laws = (id, interest, law)

6 ER Diagram

The ER diagram above shows the relationship between all of our tables.

The “user” table contains information about a users account such as their username password (which is hashed via MD5), their firstname, lastname, etc..

One user can have multiple addresses which is why there is a one to many relationship between the “user” and “address” table. Users may also have multiple interests. To avoid data duplication, the users\_and\_interests table contains both a reference to a user and a reference to an interest. This specifies one interest of the user (a user can have as many interests as they like). The interest references an interest in the “interests” table as to avoid any data duplication.

Each interest can be associated with any number of laws. A single law can affect any number of interests. Thus, a many to many relationship between “interests” and “federal\_laws” was created. The table essentially links interests to whatever laws they correspond with. Again this table only holds references (and an id) as to avoid data duplication.

The “discussion” table contains information about a particular group of threads. Each entry in the “thread” table belongs to a particular discussion (belongs\_to attribute). Any number of threads can only belong to one discussion. Threads are composed of comments. Comments are created by users whereas a user can create any number of comments (one to many relationship). A single thread can have any number of comments but each comment can only belong to one thread.