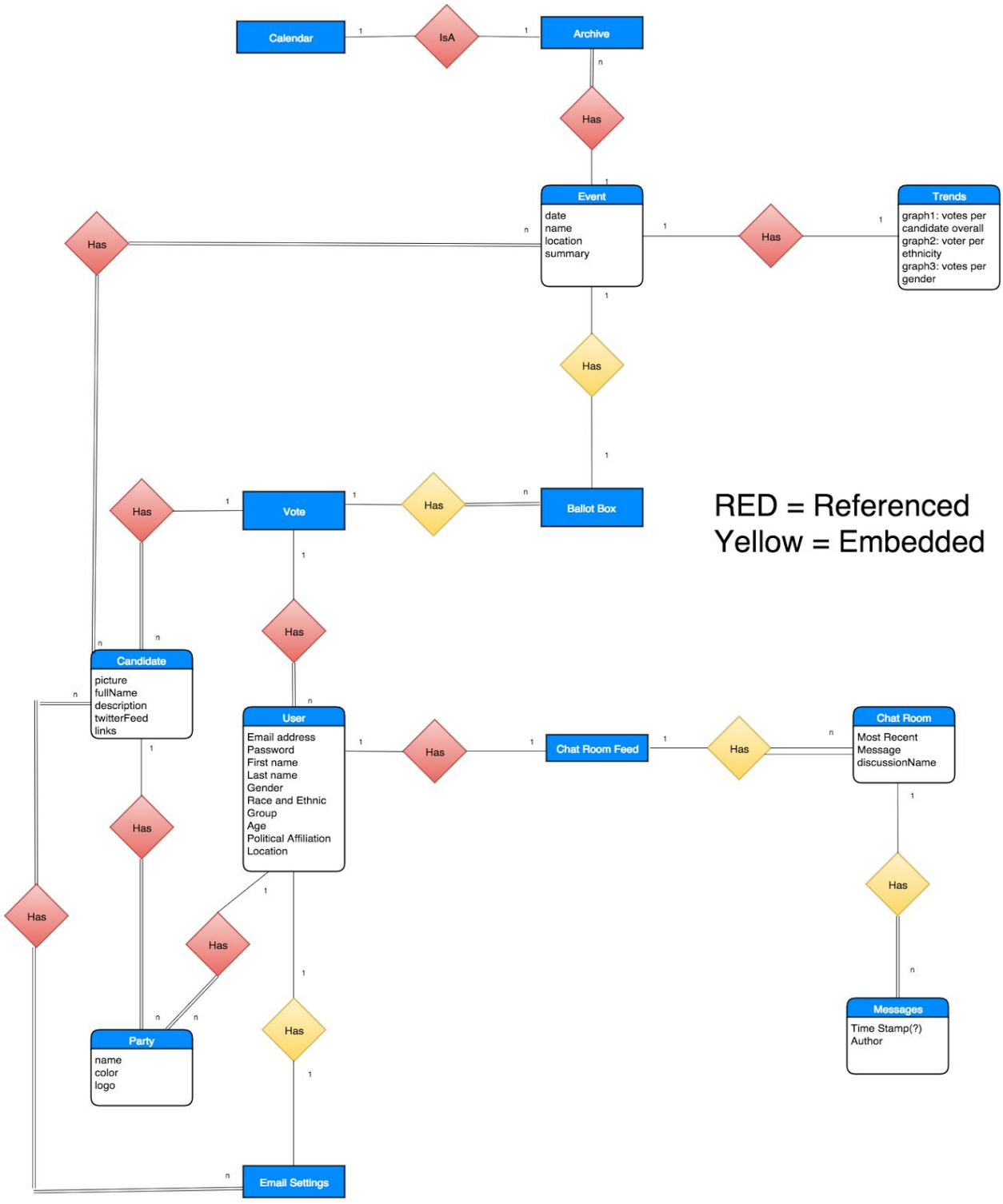
CAFEBABE's Elect.io ER Diagram

Alex Karle, Shahar Dahan, Elaina Bliss, Andrew McClusky, Neal Merkl, Alex Cobian



Our Notation:

Our notation was the same as the one used in Workshop 4. Double lines indicate one entity "has 0-n" of another entity and single lines indicate an entity "has 1" of another entity. The yellow "has" relationships indicate embedding and red relationships indicate referencing.

Entity Descriptions:

User: The User entity stores all the data that makes up the user (gender, ethnicity, age range, email… etc). The User has multiple votes, but these are referenced and not embedded because we want to access the votes without having every piece of user data attached (although each vote has a reference to a user entity which is important). Each user has their own entity. User also references a party since they have an affiliation.

Candidate: An individual that users can vote for. Candidates reference one party that they belong to and hold information about each candidate as well as a Twitter feed and links to more information for that specific candidate. The candidate has multiple votes associated with it, but each vote references one candidate. Candidates will also reference multiple events that they are associated with since their data will be used by the events.

Party: The party entity stores info specific to political parties, particularly name, logo, and color, so that each candidate can have one party reference and each party has multiple candidates. We used referencing instead of embedding because many candidates have the same party so it is easier to reference the same object. Users also reference the party since they have a party affiliation.

Vote: A vote entity does not store anything but one reference to a candidate and one reference to a user. Each user has multiple votes associated with it and each candidate will have multiple votes associated with it and each ballot box will have multiple votes associated with it. Votes are embedded within Ballot Boxes because they are small (containing just 2 references) and will not be accessed without the ballot box (embedded in an event).

Ballot Box: The Ballot Box entity will be embedded in an event because we will never access it without the event entity; The ballot box has all the votes of the users embedded in it and is essentially just a container for these vote entities. We included a ballot box instead of simply embedding votes in events because it helps with organization.

Event: The event entity represents an event in the election process which users will be able to vote on (i.e. the NH Primary). The event stores the event name, date, location, and a summary of the event. The event has an embedded ballot box entity which has all the votes associated with the event. Each event also has 1 trends entity associated with it (with the graphs) and is referenced by the 1 archive which will have multiple events. The event will reference multiple candidates involved (and each candidate will have multiple events) mostly for the purpose of the calendar display and Alex's Email Notifications.

Trends: Contains statistical information about each event and their respective graphs. Trends reference events because that is where they get their info from.

Archive: The Archive is where we store all past and future events. It contains references to all event entities, referenced instead of embedded because we want to access events without the archive (because the events are big).

Calendar: The calendar isA archive because it is a list of events and simply provides a different view of all the events.

Each Page, and How it Uses Our Entities:
Home: Has the images pulled from the candidate entities, with a highlight color pulled from the party entities. It has links to all other pages which use entities as discussed below.

User Settings: The user settings page will be how the user sets and enters their information associated with their account (gender, age range, ethnicity… etc). The page will display the current data stored in the User entity and the user will be able to update the data through using the radio buttons on the page.

Vote: The page displays who the user is currently voting for in the upcoming event (the User entity has a reference to their votes which are embedded in ballot boxes which are embedded in events. Their vote references a candidate entity which will be displayed in the UI). The page also allows the user to cast votes. This will either create a new vote if the user has not voted in the upcoming event, or this will change the candidate associated with the user's already cast vote entity.

Candidate Info: Displays the information for each candidate including their full name, picture, background info, Twitter feed to tweets about the candidate, and links to other information. This data is all stored in the candidate entity. Depending on the party that a candidate references, the page contains certain colors. This data (color) is stored in the Party entity.

Trends: This page would display all of the info gathered per event after users have voted for whoever they want. It would reach into the ballot box of each event and get the user info to populate and display graphs of the overall turnout of the event.

Archive: The archive page will access the archive entity and list all event entities that are referenced by the archive entity. It will display each event entity with some of its data (name, date, summary).

Calendar: Will use the archive entity that it is and take the events' dates and the candidates it primarily affects from the events entities and will insert them into the calendar appropriately. It will do this through the event entity info and the candidate entities referenced in the event entities.

Shahar Dahan Honor's Portion: Chat feature

The notation is the same as the rest of the submission.

Chat Room Feed: A container for the various Chat rooms. This will ideally organize the chat rooms by their most recent contained message. It will be the same for every user, so could simply be embedded in the page.

The feature would use this to organize chat rooms for the user to use.

Chat Room: A container for all of the messages regarding a certain topic. It will be the same for each user. It could optionally contain a unique chat room for those registered Democrat and for those registered Republican. Each chat room will be a new JSON object.

This would be used by the server in order to show all the messages for the users.

Messages: Each message would be embedded in its respective Chat room. It could be something like Facebook, in which only the most recent 500 or so messages are loaded unless specifically requested for more.

These would display to each user so they could communicate with other users.

<u>How it is used by the application:</u>

Every one of these pieces would be embedded in each other, as there is no reason to call any of them without the others. Without a piece of this puzzle, the other pieces would be entirely useless. They would all be called when the Chat page was opened, it's just various containers for messages.

Alex Karle's Honors Feature: Email Notifications

Our Notation:

The notation is the same as the rest of the submission.

What Was Added:

The only new entity needed for the email subscriptions is an **Email Settings** entity, which was added to the ER diagram and embedded in the User entity (explained below).

Entity Description:

Email Settings: Email Settings is embedded in the user entity, as every user has one and it will never be accessed without the user. No other entities will access it, and, as it is small, it will not burden the user entity to embed the email settings entity within the user entity. It has references to any number of candidates (from none to all), and this data will determine which candidates the user will receive emails to. The user will receive emails for events associated with each candidate entity that is referenced in the email settings entity.

How the Feature Uses the Entity:

The only page that will access the email settings entity is the Email Settings page. This page has radio buttons which allow a user to subscribe to a candidate. These buttons will add a reference from the email settings entity to that particular candidate entity. The email settings page will also use the candidate entities to display the pictures and names of the candidates and the party entities referenced by each candidate to display the proper colors for each candidate.