

Name: Anh Bui
Thanh Pham
Ngan Hoang
Ye Sung Kim
Tien Dao
Tri Nguyen

Diagram entities:

1. Users: saves information about the users like name, profile photo, email, skills, FB/LinkedIn profile links, preferred programming language, average rating,
2. Feedbacks: An entity that stores comments and ratings submitted by interviewer and interviewee after an interview session.
3. InterviewSessions: An entity that helps interviewer interact with interviewee using voice chat and code editor
4. ProblemsSolutions: An entity that hold interview questions and answers to the questions, which will be provided to users during interview sessions.
5. OnlineUsers: An entity that stores a list of users that are currently online. This list helps match the users for interview practice. When a user logs in to the Website, his/her ID is added to the list and when a user logs off from the Website, his/her ID is removed from the list.

Screens' relationship to entities:

1. Home page (home.html): Lists the user's past *InterviewSessions*, including its *date* and preview of *question* and *answer*, also include some glimpse of the feedback he/ she receives in the interview
2. Interview page (interview.html): Lists the *date*, *question*(from interviewer) and *answer*(from interviewee) of each interview. It also has *ProblemSolutions* for each *question*
3. Current User Profile (user_profile.html): Lists the user's own *email*, *full name*, *skills*, *contact information*.
4. Matching page (matching.html): A form that asks *User* to submit their interview preferences. Form submission will be recorded with *user_id*.
5. Interview History (history.html): Display a user's list of previous *InterviewSessions* on the left of the page (each interview is identified by user by its timestamp). When user click at an interview from the list on the left, the *Feedback* and *ProblemSolution* for that interview will be displayed on the right (including the basic info of interviewer and interviewee, the problem, the solution, the code produced during the interview, the comments and rating from the other user).

6. Interview Feedback (feedback.html): In the *Feedback*, it lists the *comments* and *rating* given by the interviewer and the interviewee after each interview session. When the user submits the feedback form, we will save the new feedback record with the *interviewer_id* and *interviewee_id*, in addition to the *interview_session_id*. Storing these IDs will link each feedback with the interview session, as well as the interviewer and the interviewee who participated in it.
7. Other User Profile (people_profile.html): compile a list of other users by querying the table *Users* and display basic information like name and email.

Changes:

- The interview page should be voice chat instead of the chatbox, so the chatting feature is only available in the honors section.

Honors section:

Diagram entities:

1. OnlineUsers: the same entity that keep track of online users will have another field called *location* to store the current User's location when they logged in.
2. Users: add another field to associate an user with a most recently initiated chat by this user: *chatsession_id*.
3. ChatSessions: an entity that represents different chat sessions between the users, these conversations happen in the meetup feature so that users can set up meetup themselves. It stores the *initiator_id*, *time_created*, and the *active_state* which remembers who initiated the conversation, the time it was created and whether the conversation is still active.
4. Messages: an entity that represents each *message* that an *user* sent in the ChatSessions that has *message*, *owner_id*, *chatsession_id*, *timestamp* that store the message, remember whose message it is, the session it belongs to, and the the timestamp of the message.
5. Notifications: this entity stores notifications about meetup requests that a user received, each notifications store the *from_id*, *to_id*, *chatsession_id*, *active_state* which remembers who it is from, whom it is to, the chat session that it is associated to so that we can bring the users to the correct chat session, and whether the notification is still active. (I'm not sure whether to keep track of active notifications by just adding and deleting notifications as they get created and fulfilled or not.)

Screens' relationship to entities:

1. Meet up page (meetup.html): query the *OnlineUsers* table for online users in a certain radius around the current user, and show markers on Google Map that allows the user to send meetup request. Once a request is asked to be created, the app checks for the most recent active chat session of the current user, if the session is still active then just associate the two parties to that chat sessions, but if there is no active chat session, then create a new *ChatSession* initiated by the requester. Then, create a new *notification* to the receiver.
 - Now on every page, whether the current user still has active notifications then show it up on the navbar. If the user accepts a request then add them to the *MembersList* correctly
2. Meet up chat (meetup_chat.html): now we should query the *MembersList* for *ChatSessions* the user belongs to, and list it on another navbar button so the user can access the session they want. In the chat box, the app should depend on the *OnlineUsers* entity to figure out who is online, and use the *Message* entity to save messages, and also to show past messages.