First Two Parts of Web App Overview

(Red text is newly added text compared to what I included initially)

Expectations:

- Version 0.1.x should take half a day to complete
- Version 0.2.x will probably take about 20-40 hours
- 5-10 hours will probably be spent interacting with me – walking me through the code, discussing different options when the opportunity arises (ie. AJAX long-polling vs. other methods and implications for lag and cost), and making sure we're on the same page (ie. Confirming that you understand my vision and/or your suggestions for a better way to do something)

Timeline:

- I'll send mockups either at the outset or after 0.1.x is done. UI is not very important for the scope of this project. Functionality is the most important.
- After you finish 0.1.x (if selected for this project) and push the code to Github so I can review, I'll take a look at the code and may have some questions for you. It may take me a day or two to review because I'm in a transitional period right now so I'm a bit busy with other things.
- When you start on 0.2.x, focus on getting the long polling functionality done first. We can add the other stuff iteratively. That's the most important part for me.

What you must use:

- Rails 4.0 or 4.1
- PostgreSQL
- RSpec and Capybara
- Devise and CanCan
- Heroku
- Github (I'll allow you to collaborate on an empty repository so you can push it there)
- THREADSAFE CODING!!

This application will be built iteratively. Each minor version [0.X.0] indicates an addition of a new and significant feature. I have 8 features that I would like to implement before launching, but let's just start off with this. Here are two of the planned versions:

Version 0.1.x: Authentication and Authorization via Devise and CanCan

- User registration, login, logout, forgot password, allow user to enable two-factor authentication, welcome email, confirmation email
- Authorization. Should have these roles (features of each role listed in bullet points below will be added in a later minor version that is not within the scope of version 0.1.x. This is just to give you an idea of what they can do):
    - Super Admin
        - Super admin can appoint and change roles/permissions for any user
        - Can view a list of all users, their firm, and their roles (including super admins)
    - Sales Manager
        - Has the same features/permissions as sales admin and sales host
        - Additionally, the sales manager can view analytics (analytics to be added in a future version)
    - Sales Admin
        - Has the same features/permissions as a sales host
        - Additionally…
            - Can appoint/revoke roles for any sales host, client admin, or client user
            - Can schedule, change, or cancel any webinar
    - Sales Host
        - Can schedule and generate a webinar session
        - Can host a webinar session

        - Can dynamically generate documents from a template
    - Sales Compliance
        - Can dynamically generate documents from a template
        - Can generate reports (logs, analytics reports, etc.)
        - Can set up regular reporting sent via email
    - Client Admin
        - Can view analytics for client admin's firm only (cannot view analytics for other firms)
        - Can view usage logs of client users in this client admin's firm
        - Can join webinars that are open client admin's firm
        - Can view results of continuing education modules completed by client admin's firm's client users
    - Client User
        - Can take continuing education modules
        - Can join webinars open to this client user's firm and webinars by invitation
- Do not allow the same user to be logged in on two different devices simultaneously. The 2nd device that tries to log in can kick off the first device; the first device should be notified that a 2nd device kicked him/her off. Flag this activity to Super Admin.
    - This is to prevent two people from participating in a webinar or perusing the site without being with each other in the same location on the same device.

Version 0.2.x: Webinar Presentations

- Sales host can advance slides (or go back to previous slides) and use actions (sequenced animation) and/or change variables on a slide (ie. Changing an input to change a bar chart). Any changes on the host's screen will appear simultaneously on client's screen. Must minimize lag.
    - Use AJAX long polling or Rails stream/live
        - I may consider using socket.io and Node.JS but I would prefer not to because of my lack of experience with that technology.
    - Using websockets alone won't be sufficient because it isn't supported by IE 8 & 9
    - Also needs to work for Android browser version 4 and up
    - To keep costs low, I'm looking for function and clean DRY code with good tests written in RSpec/Capybara. I don't need a lot of work on the UI for this long polling feature because it will likely change as I iterate outside the scope of this initial project
        - Essentially, I want you to set up a barebones working infrastructure with good tests. Then I want you to walk me through it so I can expand it with content and additional features.
    - Code with the consideration of minimizing costs and lag. I don't want to throw resources at it when better architecture can solve the problem
    - CODE MUST BE THREADSAFE
- The sales host's screen should have a listing of all attendees currently in the webinar.
- Sales host can create different types of webinars:
    - Open to any logged-in user
    - Open to any logged-in user from whitelisted firms for this particular webinar
        - This will require a feature that allows a sales host to add/remove firms in the DB
        - When scheduling a webinar, the host can select from the list of firms in the DB to be allowed to participate in the webinar.
        - When a user is logged in to participate in the webinar via clicking on a webinar-specific URL, the app checks to see if the user belongs to a firm that was whitelisted for this particular webinar. If true, then user joins webinar; else user is redirected to a page that says that the webinar is not open to his/her firm and to ask them to contact the sales firm.
    - Invited users only

- Similar to above except app checks if user is on invited list instead of whitelisted firm list.
- Invited users may not be registered before being sent an invite. However, the sales host should be able to input the email address of the client user. The app will match emails of the invite list vs user's email to see if he/she was invited.
- This means that the sales host can search for users by name, email, firm, city, state to place them on the invited list. The sales host can also add by email if they're not in the db (because they haven't signed up yet)

- Each webinar should have a unique, dynamically generated URL which can be sent in email invitations
  - The implication is that the URL may be generated several weeks in advance.
    - If a user clicks on the link before the session starts and logs in, the page will say "This webinar is scheduled for [date and time slot]"
    - If a user clicks on the link 10 minutes before the session AND the sales host did not officially start the webinar AND user logs in, then the page will show "This webinar will start in approximately [time until scheduled start] minutes"
    - If a user clicks on the link during the session AND the sales host did not officially start the webinar AND user logs in, then the page will show "This webinar will start shortly."
    - If a user clicks on the link during the session AND the sales host started the webinar AND user logs in, then the user starts off with whatever slide the host is on. The user does not start at the beginning of the presentation if the sales host is already beyond that slide.
    - If a user clicks on the link after the webinar ended and logs in, then the page will show "This webinar has already ended. A replay may be available. Please call [sales firm's contact number] or email [sales firms' email contact] for a replay or request a private webinar."