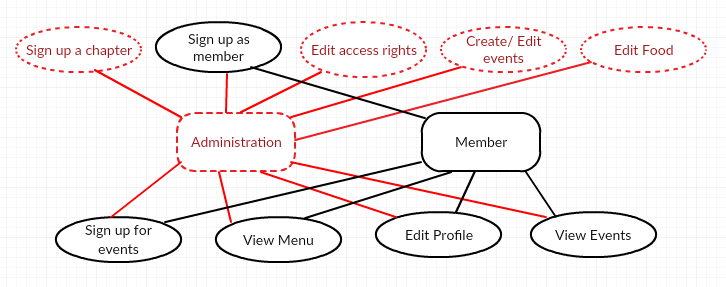
Jennifer Barry

Meredith Weiner

Advisor: Professor Femister

**Project Design**

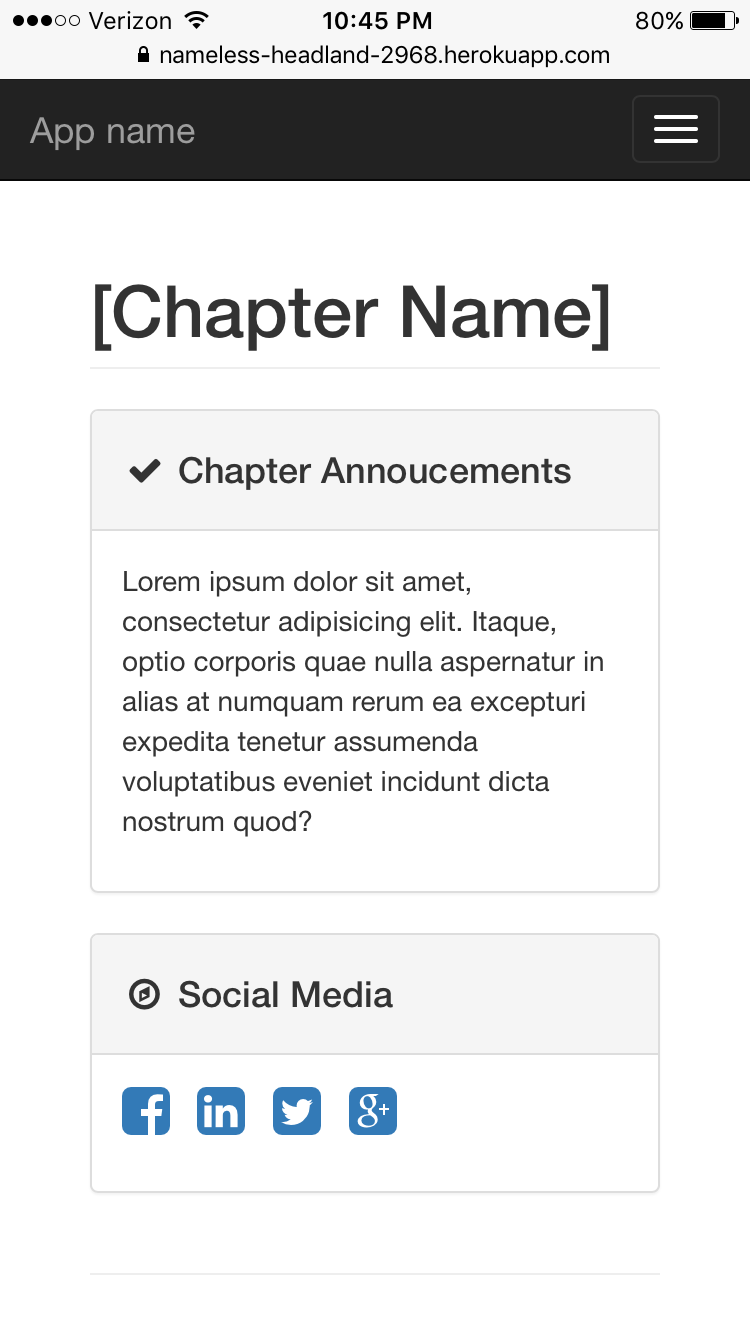
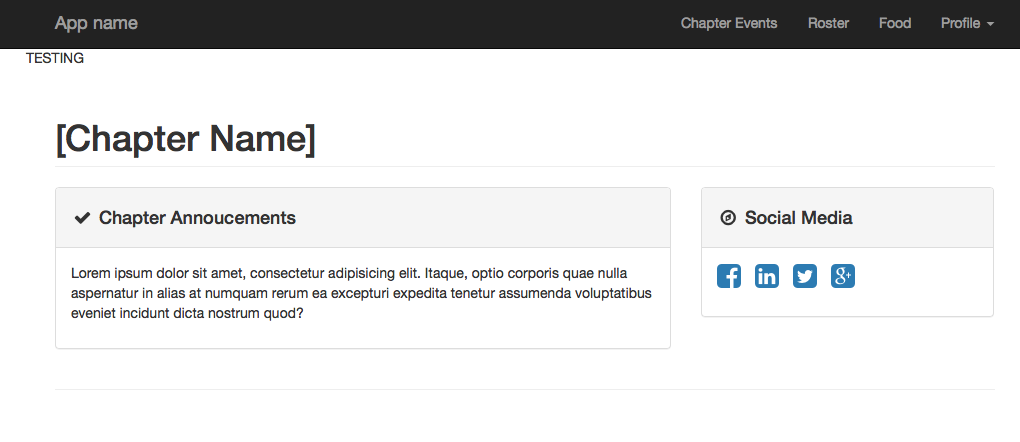
**Objects and Interactions**

****

The above diagram displays the roles of members with administrative rights versus general members. Administration will be given additional rights because they need to have the authority to make the necessary changes. Everything that a general member can do can be done by administration as well. The black parts of the diagram show the interactions for a general member, which includes signing up as a member, signing up for events, viewing the menu, editing their own profile, and viewing events. The red parts of the diagram correspond to the roles of administration which additionally include signing up the chapter, editing access rights for members, create, edit, and delete events, and editing the menu.

**Responsiveness:**

When designing our application, we always keep the clients in mind. We know that our users aren’t sitting on their computers all day so our application must work on mobile devices as well as on a desktop. Originally we planned on implementing mobile development after the desktop development, but we realized that wasn’t practical. We decided to instead develop our desktop version to work on mobile phone concurrently. As we implement features, we test them on mobile devices as well to ensure that everything is compatible across platforms. An example of our homepage is below, one version from a laptop computer and another from a mobile device.



**Keep the Target User in Mind**

One of the main features of our application is the calendar, which administration can create, edit, and delete events and general members can sign up for events. Originally, we thought we were going to embed Google Calendar into our application but after extensive research, we realized that was not going to be feasible. While the Google Calendar API lets you display, create, and modify calendar events, there is not much flexibility to modify these built-in features to cater to our needs.

In order to ensure that we keep the target user in mind and incorporate the necessary features, we have decided to use Node.js, which would enable us to write code to build a customized calendar. This will allow us to have members view an event on the calendar and sign up using a form that we create. We want everything to be done within our application, as opposed to clicking on a link, which brings the user to an outside source, such as Google Calendar. This is why we have strayed away from this technique.

Our main concern is our user, and we want our application to be as convenient for them as possible. We feel that the best way to achieve this is through the implementation of our own customized calendar.

**Security:**

We know that security is an important issue. All of the data contained within our application is internal chapter information. Due to this, we do not want anyone signing up without being a recognized chapter member. In order to prevent unauthorized users, we have created a verification system. Before any members can register, their chapter must be signed up. During this process, the chapter president must provide a roster of all active chapter members. Then when chapter members attempt to register, they will be verified against this roster to ensure they are members. They will also need to enter a code provided from the president that ensures they are the person they are claiming to be online. This will allow chapters to use our application to communicate nonpublic information to their members in a secure manner.

**Progress:**

We are currently on track with our application based on our original Gantt chart. While we planned on completing many of the tasks independently, we realized that concurrent development of some of the features was the most efficient use of our time due to the time constraints we have been facing. As a result, we have begun integrating our database along with our front-end features. Neither of us has taken a database course so integrating our database early on allows us to gain experience before our application gets more complex. While some of our front-end features will be developed later on in the semester, we began using our database weeks earlier than originally intended. This puts us in a good position for the rest of the semester.