TEAM SPIDER CODE CRITIQUE

By Team ProjectGroup

Yixin Li, Yihua Li, Michael Wee, Faith Keza

SimpleTime and SimpleDay

- Are these necessary? Maybe you could use Ruby's built-in Date and Time classes and methods to achieve the same result.

- Not all of the methods in these classes are used throughout the application.

- For example, good amount of comments in SimpleDay and SimpleTime so that the reader understands what the class represents. Another good comment example is specify what the parameters the partial uses in views/layouts/form_errors_html_erb.

SimpleTime and Simpleday Invariants

- SimpleDay seems to have the invariant that 0<=day < 7, but how does "succ" method wrap around Sunday if the value is already 7?

- SimpleTime seems to have invariants that 0<=hour < 24 and 0 <= minute < 60. The private method "fix_minutes" makes sure @minutes is in the range [0, 60], but how does it handle the case when the @minute become negative or when @hour becomes larger than 24?
  - (For example, SimpleTime(1,10)-SimpleTime(1,20)).

Formatting and Style

- README file is empty. Since it is the landing page of your repository, it really needs to help the viewer navigate your files and documents.
- There are empty files in your application. For example, there are several of the files in app/views/location.
- Right amount of comments in many places. For example, good amount of comments in SimpleDay and SimpleTime so that the reader understands what the class represents. Another good example is specifying the parameters that the partial uses in views/layouts/form_errors_html_erb. The code in app/views/devise, however, might need more comments.
- Great work with fat model/skinny controller, however, be careful with some queries in the your controllers that can be computed in the methods in the model.
- Good separation of concerns -- you have no controller logic in your views, and no inline CSS/JavaScript in your HTML.
- The use of partials make the code very concise.