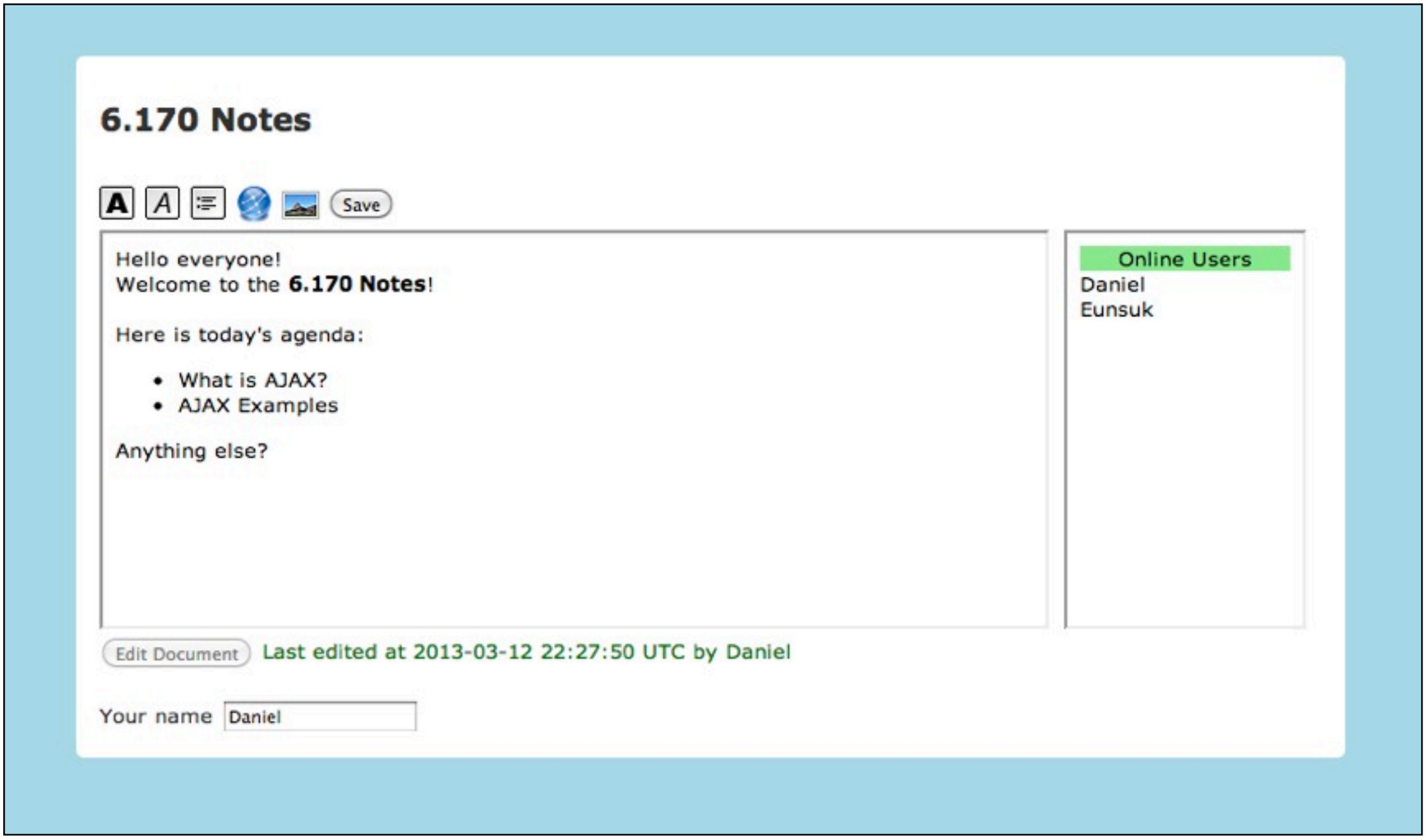
software studio

asynchronous calls: Engage app

Eunsuk Kang

challenge



- multiple viewers
- update view when document changes?

design alternatives

users manual refresh

› easy, but undesirable

polling

› client sends a request periodically
› simple to implement
› scalability issues (wasted requests)

server-side push

› server sends out updates to clients
› efficient, scalable
› no built-in support in Rails
› Juggernaut, Comet, WebSocket, etc.

polling

```
# Return the latest version of the document for this project
# GET /latest_doc
def latest_doc

  project = Project.find(params[:id])
  doc = project.extract_latest_doc

  render :json => doc
end
```

server side

exercise: can you come up with the client code to perform polling?

- assume JS function **update_document(doc)**
- also available: **$.get**, **$.ajax**, **setInterval**

polling

```ruby
  # Return the latest version of the document for this project
  def latest_doc
    project = Project.find(params[:id])
    doc = project.extract_latest_doc

    render :json => doc
  end
```

server side

```
var POLL_INTERVAL = 5000;

var poll = function () {
  $.ajax({ url : '/projects/<%= @project.id %>/latest_doc',
           success : function(response) {
             update_document(response);
           },
           type : 'GET', dataType : 'json'});
};

var start_polling = function () {
  setInterval(poll, POLL_INTERVAL);
};
```

client side

› model instance automatically converted into json

custom JSON in Rails

```ruby
# Return the latest version of the doc and
#        the set of users who are currently online
def status
  project = Project.find(params[:id])
  json_response = { :document => project.extract_latest_doc,
                    :users => project.online_users }

  render :json => json_response
end
```

server side

```javascript
var poll = function () {
  $.ajax({ url : '/projects/<%= @project.id %>/status',
           success : function(response) {
             update_document(response.document);
             update_online_users(response.users);
           },
           type : 'GET', dataType : 'json'});
};
```

client side

AJAX summary

in Javascript

- timers
- jquery methods

in Rails

- constructing JSON
- form helpers

6.170 Software Studio
Spring 2013