Kyle Garber

CS 491

**CS 491 Engineering Notebook**.

**1/17:**

* Team Meeting:
  + Reviewed status of the project
  + Arranged sub teams
    - Carly and I are predominately on UI, with my focus being the FO interface

**1/19:**

* Team Meeting:
  + Reviewed upcoming vision statement and backlog assignments
  + Visualized project updates based on UI design proposals Jesse created
  + Begun work on backlog items in sub teams
* Begun work on combining the fo and tc login pages by moving them to the home Django app

**1/20:**

* Spent time considering current design and possible directions for the project since customer provided availability for meeting

**1/22:**

* Continued design contemplation as customer meeting had been scheduled for 1/26

**1/24:**

* Team Meeting:
  + Juliana shared new information from customer regarding project goals
  + Team discussion on design implementations to achieve new goals
* Continued working on backlog items for FO UI, Documentation and Planning, FD UI, and Communications epics
* Review of revised mission statement for the project that more accurately reflects current designs and goals

**1/25:**

* Prepared questions for customer meeting tomorrow, 1/26
* Begun work on merger of the fo and tc login pages

**1/26:**

* Team Meeting:
  + Met with customer
    - Discussed mission length
    - Discussed mission design
    - Discussed control design
    - Discussed team design
  + Digested customer design changes as a team

**1/27:**

* Finalized merger of fo and tc login pages
  + All signup and login is now handled by the home django app
  + fo and tc users are then directed to their app based on user type

**1/30:**

* Researched ways of asynchronously updating page content and database values
  + Ajax appears to be the most popular solution

**1/31:**

* Team Meeting:
  + Continued discussing customer changes
  + Debated value of team over single flight director administration of a SimCraft
  + Received official task for handling Ajax implementation
* Continued research into Ajax

**2/1:**

* Created new models in fo.models to test out Ajax database manipulation
* Created new urls to host and handle Ajax requests and tests

**2/2:**

* Team Meeting:
  + Shared progress on Ajax prototyping
  + Discussed individual subsystem designs
* Registered new models in fo.admin to monitor values on admin page

**2/3:**

* Imbedded Ajax script into foHome.html to handle a button press

**2/6:**

* Imbedded more Ajax functions into foSim.html:
  + Handle press of a submit button
  + Execute periodically to update display objects

**2/7:**

* Team meeting:
  + Further discussed Ajax implementation and project design

**2/8:**

* Created handlers in fo.views that manipulate database objects when called by Ajax scripts

**2/9:**

* Team meeting:
  + Discussed Ajax progress and committed to including Ajax functionality in upcoming demo
* Changed database fetching of objects from name handle to pk handle for all objects without mandatory unique names

**2/10:**

* Passed arguments from Ajax functions to functions in fo.views to determine new values for database objects being manipulated by the view function called
* Experimented with success functions to be executed by Ajax after fo.views function returns.

**2/11:**

* Successfully passed data from frontend foSim.html web-page to fo.views, manipulated database values, fetched new values, and returned new values to foSim.html to be displayed without web-page reload
* Functionality of previous demo using django forms has now been completely replaced and improved upon via the use of Ajax

**2/12:**

* Cleaned Ajax calls and foSim.html for maintainability
  + Relocated javascript to separate files referenced with script tags
* Committed and handled conflicts for Ajax functionality

**2/13:**

* Directly imported jquery.min.js to library to reduce reliance on third party hosted utilities
* Removed deprecated models and urls used for testing Ajax

**2/14:**

* Worked on presentation slides
* Prototyped generic script to be executed from the tc django apps views.py script
* Prototyped database manipulation via script external to a django app
* Team Meeting:
  + Presented for Inc 2. Sprint 1
    - Demonstrated current Ajax functionality
  + Discussed simulation script starting conditions

**2/15:**

* Prototyped new django app simapp to take over sim database functionality currently handled by tc
* Experimented with new input types on foSim.html

**2/16:**

* Team Meeting:
  + Discussed simulation script-starting
  + Discussed subsystem database design
  + Discussed fo input types to be hosted in foSim.html
* Added Ajax handling to new input types
* Finalized and committed new simapp django app
  + Changed references to tc.models sim functionality to simapp.models
  + Reset database and handled merge conflicts
* Renamed SWA/Simulation to SWA/simulation to follow naming conventions