Kyle Garber

CS 490

**CS 490 Engineering Notebook**

**9/13:**

* First team meeting with product owner (Professor Garfield) and customer (Professor Laskey).
* Established product vision and expectations.

**9/14:**

* Researched potential app development frameworks.
* Created survey for team to vote on app development framework to be used.
* Created getting started epic and added initial tasks.
* Created sprints in zenhub.

**9/15:**

* Tallied survey results and picked .NET Maui as our development framework.
* Created initial .NET Maui project to be used by team.
* Added .NET Maui project to team’s GitHub in /src folder.
* Researched and experimented with git and .NET Maui app.
* Had meeting to further develop team vision for the application.
* Added 9 new epics to zenhub.
* Created 14 new backlog items on zenhub in new epics.

**9/19:**

* Created images for use in .NET Maui application.
* Drafted entry page with options for student and professor login.
* Drafted student login page with options to return to entry page or enter session code.
* Drafted professor login page with options to return to entry page or enter password.

**9/20:**

* Worked on draft entry, student login, and professor login pages.
* Met with team to discuss project direction.
* Researched alternative methods of project development and compiled findings into a shared team document.

**9/21:**

* Added more development research/thoughts into shared team document.

**10/4:**

* Team Meeting:
  + Discussed and assigned documentation work.
  + Discussed demo plans and goals.
  + Split into sub-teams for webapp and simulation work (I am on webapp).
  + Discussed sprint 2 goals.
* Experimented further with web design templates.

**10/10:**

* Team Meeting:
  + Met with Professor Garfield and was provided with more system details.
  + Product description is on the way.
* More focused research on product development using Prof. Garfield provided details.

**10/12:**

* Begun development on webapp login page.
* Experimented with databases, namely mongoDB, and linking mongoDB to webapp to store user information.
* Researched NASA spacecraft console design.

**10/13:**

* Team Meeting:
  + Discussed product description provided by Professor Garfield.
  + Begun development of sprint one demo presentation.
* Worked more on webapp login page and website routes.
* Researched similar applications and their respective frameworks.

**10/15:**

* Started work on a high-level UML diagram for presentation and webapp development.
* Considered several different software design options.

**10/16:**

* Finished high-level UML and added it to presentation.
* Removed deprecated folders from git repository.

**10/17:**

* Created hardware diagram that shows relationship between hardware necessary to our project and added it to sprint one presentation.
* Completed several sections on the sprint one presentation.

**10/18:**

* Finalized sprint one presentation and rehearsed sections assigned to me.
* Presented sprint one presentation alongside the team.

**10/24:**

* Researched alternative web application frameworks and databases since I was unhappy with the node js / mongo DB structure we had chosen.
* Decided the Django web application framework would be preferable for our project.

**10/25:**

* Shared thoughts on switching frameworks and discussed potential alternatives.
* Decided with team that Django would be preferable.
* Discussed implementation options with the team and began mapping out project structure within the Django framework.

**10/27:**

* Created the initial STaTE Django project.
* Added testapp to the Django project, a Django application that explores several features of the Django framework and can be used by developers to test ideas.
* Researched Django implementation strategies and common project structures.

**10/28:**

* Added features to testapp in similar fashion to the official Django website tutorial on creating the polls app.

**10/29:**

* Initialize section 4.1 of the STaTE SRS document pertaining to the functional description and requirements related to SWA (STaTE Web Application).
* Expanded on the functional requirements of the SWA and added requirements related to maintained URLs and site page navigation.

**10/30:**

* Changed name of STaTE Django project to SWA to reflect terminology used in the SRS.
* Added fo, tc, and home apps to the Django project.

**10/31:**

* Committed SWA name changes.
* Implemented basic url navigation among pages defined in the SRS via html button elements.
* Added page identifier messages for testers to verify the correct page is being displayed.
* Experimented with html embedded logic in testapp Django app.

**11/1:**

* Initialized and committed system test plan document.
* Finalized and committed basic navigation changes between SRS defined pages.
* Team Meeting:
  + Discussed requirements for subsystems in the SRS that will have to be completed before finishing the test plan.
  + Discussed plan for finalizing and submitting test plan.

**11/2:**

* Completed section 1 of the test plan.
* Completed section 4 of the test plan including the execution plan for the SWA subsystem functional requirements.

**11/3:**

* Team Meeting:
  + Led team meeting discussion about url routes and Django web application view design.
  + Discussed user account methodology and authentication requirements.
* Begun incorporating Django user management and authentication into fo app.

**11/4:**

* Expanded on user management and authentication for the fo app by implementing infrastructure to create FlightOperator users through browser input.
* Migrated changes to FlightOperator user model to the database.

**11/5:**

* Implemented pages to allow for user login and logout within the fo Django app.
* Tested communication between Django apps by displaying an authenticated fo FlightOperator’s username on the home page within the home Django app.

**11/6:**

* Fixed Django navigation issues related to navigation from fo pages to testapp and tc pages.
* Verified fo users were not given administrator privileges when attempting to access admin Django page.
* Added authentication logic to verify user is signed in to view pages beyond the fo login page.

**11/7:**

* Updated section 1 in the SDDv2 group document.
* Created a use case diagram for the STaTE system.
* Researched constraints associated with the chosen design of the STaTE system.

**11/8:**

* Team Meeting:
  + Discussed system architecture.
  + Discussed methodology of internal communications for various subsytems.
  + Assigned parts of the SDDv2 to be completed.
* Updated UML to account for architecture and communication changes.
* Added UML and subsystem descriptions to section 4.2 of the SDD.
* Finished section 1 of the SDD.

**11/9:**

* Added SDDv2, SRSv2, and TestPlan documents to github.

**11/10:**

* Team meeting
  + Further discussion on user interface layout
* Formatting/style changes to the top-level SWA application

**11/11:**

* Researched Django models and ways to implement them
* Researched Django views and methods of loading and rendering html templates from the Django backend

**11/12:**

* Added crispy\_forms module to project for form generation and rendering in foLogin.html
* Refactored fo login page, fo/views,py, and fo/urls.py to streamline login for flight operators

**11/13:**

* Added foHome, logout, and register paths to fo/urls.py
* Added createSim url path to tc/urls.py
* Changed index to home url in home/urls.py

**11/14:**

* Added simulation() model to tc/models.py with testArgument and sim attributes
* Added FlightOperator() model to fo/models.py with user and email attributes

**11/15:**

* Team meeting:
  + Discussed team subgroups more and assigned teams of two to various pieces of product functionality
  + I was assigned to lead top-level app (home Django app) development and assist with fo app development
* Experimented with email backend in settings.py
* Added LOGIN\_REDIRECT\_URL to settings.py
* Cleaned up and committed changes from the last week to main github branch

**11/16:**

* Added html templates for the fo Django application
  + fo/foHome.html
  + fo/index.html
  + fo/register.html
* Added tc/createSim html template for the tc Django application
* Added DjangoFilmsCRUD-master Django project to src for team to use as a reference
* Cleaned up github folder names with spaces. Replaced with underscores

**11/17:**

* Team meeting:
  + Discussed product layout and style implementation.
* Begun researching methods of styling html objects with css

**11/18:**

* Researched how to set background of html body objects to an image
* Looked for suitable images of stars to use as the background for SWA home and login pages

**11/20:**

* Modified home Django app index page to match style used by fo login and registration pages
* Added upper navigation bar to home Django app index page

**11/21:**

* Added static folder to Django project to hold static css and javascript files
* Added index.css to serve as main local repository for the apps css classes
* Added media folder to Django project to hold image and video files
* Registered static and media folder with the Django project in settings.py
* Set the background of home Django app pages and fo/tc login and registration pages to start.png in the new media folder

**11/22:**

* Team meeting:
  + Assisted team in fixing problem with launching Django project (crispy-forms module was not downloaded)
  + Discussed implementation and handling of forms for user input
* Researched Django requirements.txt files and how to implement one to automatically download required modules before launching the SWA Django project

**11/23:**

* Updated all Django app’s urls.py and views.py files to better handle user navigation with consistent methods
* Updated and registered fo app FlightOperator model
* Performed database migrations to update database with most recent model templates
* Updated html templates to user new naming conventions in all app’s urls.py files

**11/26:**

* Researched new Django email interfacing methods as the previously used method is unreliable
* Created [state.notification@gmail.com](mailto:state.notification@gmail.com) account
* Altered [state.notification@gmail.com](mailto:state.notification@gmail.com) account settings to allow for external entities to sign in via two factor authentication

**11/27:**

* Updated user registration form to submit user first and last names to the User object when and account is created
* Researched and experimented with ways to create side navigation bar on foHome page
  + Div html element with 100% height and 25% width seems to be the ideal option

**11/28:**

* Updated style and html elements on home and fo Django app top-level pages
* Implemented new email account usage and updated settings.py to hold email account and login information

**11/29:**

* Team meeting:
  + Fixed Django deployment issues with teammates
  + Discussed what will be able to get done by the end of the semester
  + Decided that a meeting with Product Owner and (hopefully) customer should be made a priority before the semester is over
  + Begun discussion expectations for project work over winter break
* Added button styles for the Flight Operator and Test Conductor login buttons on the home app’s home.html page
* More url and html changes
* Researched methods to standardize url implementation in a Django project
  + Name-spacing appears to be what we need to maintain consistency across our Django apps

**11/30:**

* Migrated database changes and committed updated database with several default accounts

**12/1:**

* Team meeting:
  + Discussed Sim implementation
  + Discussed default subsystem properties
  + Discussed methods of creating custom subsystems
  + Obtained meeting with Product Owner for next Tuesday (12/6)
  + Discussed possible demonstration structure to present to product owner
* Begun creating database infrastructure to be used in demonstration
* Name-spaced fo app for consistent url naming conventions

**12/2:**

* Implemented sim-list on foHome page that displays all of a FlightOperator’s active sims as links
* Implemented sim page that uses the selected sim’s name as a url (fo/simname)
* Updated sim page to show each of a sim’s subsystems as a div with the sims name
* Updated sim divs to hold a submittable form that allows for sim attribute modification in the database
* Added models to both fo and tc models.py files to support necessary database operations
* Added form templates to fo/forms.py to be used in subsystem divs

**12/3:**