Plan for Texas Holdem Web App

**Goal:** To update and expand functionality of existing Django app.

1. Update front end technologies to utilize the modern frameworks of AngularJS and Bootstrap.
2. Incorporate User Authentication into app
3. Display current matplotlib charts and tables in Web App using png files, and where appropriate convert more modern frameworks like Bokeh, mpld3, and D3.
4. Setup App to run on Apache Server.
5. Expand capabilities to calculate probabilities for the flop, and turn.
6. Develop a game app where players can login and join a table and play a game in real time.

**Updating Front End Technologies:** Currently the App utilizes a base template using the Django templating language where a sidebar on the left side of the screen which contains the Navigation links. The rest of the page for the main content. The seven navigational links are:

1. Create Simulation - This page creates a simulation job based upon a user input form. The form requires: the number of players, and 4 simulation parameters (the number of CPUs, the number of games per simulation loop, number of simulation loops, and the directory to output data). These parameters are used to adjust the load of the app on a particular server. Finally there is an option to save all of the result game data not just a summary.
2. Simulation Queue – This page displays a dynamic list of jobs in to queue. Each job list is linked via the REST framework to the database, giving the user CRUD capabilities. In addition on this page also contains the Run Jobs button which submits all of the jobs in the queue.
3. Simulation Dispatcher – This page shows the CPU statuses of the current job running and the time which the job is running. The dispatcher will sequentially run all jobs until the queue is empty. There is also a button to rest the dispatcher if required.
4. Create Analysis – This page create allows to user to create an analysis job based up the number of players, and a date range.
5. Analysis Queue – This page displays all of the successfully finished simulation jobs for the date range selected. The user has a set of check boxes to select which jobs to be included in the analysis.
6. Analysis Dispatcher – This page displays the status as each step in the analysis is completed.
7. Initialize Simulation - This page is an administration page which initializes all of the necessary lists and dictionaries required for the simulation.

The Django templating language includes the {% verbatim %} {% end verbatim %} tags. These tags are used to allocate an area on the page to use other templating languages like AngularJS. Currently none of the modern features of AngularJS are being use. Bootstrap integrates seamlessly into Django, and its links and scripts are already included (but only being used minimally). The date pickers from the jQuery interface are being utilized. The front end will also be changed to incorporate front end routes using angular ui router.

**User Authentication:** Currently the App does not utilize user authentication. Since the database has already been setup in MySQL, and Django automatically setup the authentication system, only the python code to access these database tables and the login view need to be written.

**Displaying Current Analysis:** Currently the analysis portion of the App includes a fairly complete analysis of hole hands (12 graphs and tables), that is the cards which are unique to a single player and are not shared with the other players. The charts and tables are currently being save as png files and not being displayed. At the very, least these files will be dynamically rendered as is using the image tags but where possible these images will be rendered using Bokeh (or mpld3) and D3 to give the user the best experience of being able to pan, zoom, and save the images from the front end. Research will need to be done to determine if it is possible to rotate the 3-d views in the browser or if these images will just be displayed statically.

**Setup and Run App on Apache Server:** Currently the app runs on the Django development server. The app would be transferred over to an Apache server either installed locally or in the cloud.

**Expand Analysis to Calculate Probabilities on the Flop, and Turn:** The same probability algorithm can be modified to calculate the probabilities at other points in the game. A set of views would be setup to calculate to probability of each player winning for a particular set of inputs (hole cards, flop cards, and or turn card.

**Expand App to Play Game in Real Time on the Web:** Players will be able to join a live game and play a game across the Web.