Angular App Planning Template

Guidelines:

Start by writing apps purpose, and main components. Components should be nested and refer mainly to big UI pages/elements, with smaller re-usable components described as well.

Then move to data MODELS. Define each model, the variables it will hold and it’s methods. Also describe which components have access to them.

Then move to services, similar in some ways to models. Describe the functions and the components which will have access to them.

Directives; Outline any custom directives and their purpose only after searching for a third party solution. UI

# App Purpose

The purpose of the app Is to serve as something of a message board in which users can suggest an activity that an entire community can get behind, pledging funding… to be done on a specific day. On the day either the funding provided is refunded to the community contributors if the task is not successfully accomplished, OR it is awarded to the ‘poster’ of the challenge.

# Components

**App Component –** Holds APP frame

**SignIN/OUT Component –** Login/Logout functionality

**UserComponent –** User data including current wealth, challenges, participations.

**Upcomming challenges Component –** Shows upcoming challenges for this week as per current votes.

**Yesterday’s challenge Component –** Is a blog of the previous challenge and how it went.

**Today’s challenge Component –** Today’s challenge, with comments section and live blog/chat

**Comment Panel Component -** Component which shows individual component elements.

**Report Panel Component –** Reports messages

**Tommorow’s Challenge Component –** Interface to see and create challenges for a future date.

**Challenge Response Component –** Component which shows individual challenge elements.

**Report Panel Component –** Report challenges.

**ChallengeFullView Component –** shows full challenge data including comments for individual challenges, allows early donations etc.

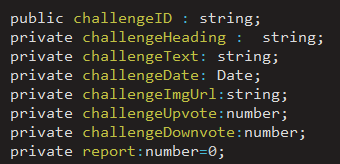
**HomeScreen Component -** Homescreen

**Archive Component -** Archive

**Page Not Found Component –** Page not found message.

# Models

### Challenge Model

**-**

Also needs:

-donations

-user who created it

-Challenge date created and date challenge is set at.

Functions:

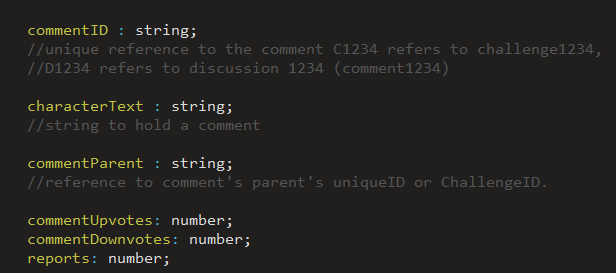
-Getters/Setters

-upvote/downvote/report incremeneters.

-edit challenge.

-attach media.

### Comment Model



Also needs:

-user who created it

-Date created

Functions:

-Getters/setters

-Upvote/downvote incrementers

-Edit

### Donation Model

-Donation Target (which challenge it’s donated to)

-donation username

-donation amount

-donation date

-backup donation refund method/address

-donate Status

Functions

-donate

-edit donation

### User Model

**-**Username

-Password

-Recovery Question

-email

-Recovery Answer

-challenges posted

-comments made

-donation pool

-donations active.

-account status

-reported content

Functions:

-change user status

-change user email

-change user password/recovery questions

-add challenges posted

-add comments made

-add donation pool

-subtract donation pool

-add donations active

-subtract donations active

-upgrade account status

-downgrade account status

-add accountReport

-downgrade accountReport.

# Services

### - HTTP service (responsible for HTTP CALLS)

-Challenge Service

Responsible for building challenge lists from HTTP calls, and disitrbuting them as required.

Also responsible as a middle-man for adjusting individual challenges, and uploading them back to the server.

-Comment service

Responsible for building comments from HTTP calls, and disitrbuting them as required.

Also responsible as a middle-man for adjusting individual comments, and uploading them back to the server, and also deleting them etc.

-Auth Service

-Responsible for authenticating users, and handling routing around authentication. (preventing protected areas of the site from non-authenticated users).

-responsible for login/logout functionality

-changing recovery questions etc, gets routed through here.

-User Service

-Responsible for serving user data, responsible for filtering users based on categories and returning lists of users.

-Calls HTTP method to alter any user elements.

# Directives

# To do list

* Databse linking
  + Plan full database interactivity
    - Replicate what you’ve done for challenges to comments.
  + Code full database interactivity
* Homescreen component
* Sign In / Sign Out functionality
* User component design
* Upcomming challenges page
* Sort comments frontend
* Bugfix sort doesn’t work on all of them for some reason, button only swaps between hated and top. Going top->bot doesn’t work or something similar.

Fix models to include all necessary data