Team Tsunade

**Project ClassSync**

\*This document describes our team social contract for this project\*

Reference Establishing Norms

Roles

* Taskmaster/Gitmaster rotating weekly
* Taskmaster responsible for directing stand-up meetings and supervising task assignments
* Gitmaster responsible for curating task board and overseeing pull requests/merges

Daily process

* Morning standup 10:45AM/after REACTO, 2:30PM check-in, 5:30 merge meet
* Standup Process
  + We individually discuss progress from previous day and goals
  + Decide if we need to continue existing pairs to resolve previous blockers
  + Set up issues for the day
  + Breakout in groups
  + Set up Issues w/ # for the day for the two teams
  + Yesterday? Today? Obstacles?
  + At 2:30pm have re-group meeting where you check-in with blockers, status updates and
  + EOD meeting is dedicated mostly to reviewing pull requests from the day’s work
* Pair program on assigned tasks
  + Taskmaster to assign groups
  + Group discussion on assignment of task
  + Teams to decide their daily goals and milestones
  + Teams will be responsible for one micro task in between standups
  + Notify the other team of early completion- if second team is not done then check with second team and  move to or selects second task to begin i
  + If the second team is stuck  then Mob code after standUp
* Record issues that are non-blocking and discuss right after lunch at re-group meeting
* Bring up any blocking issues after 15 minutes of spinning your wheels

DEVOPS- Meeting

* Plan for EOD merge 5:30 PM  each day - Discuss what was completed and plan for tomorrow
* Contribution Process Review:
  + Make an issue (or multiple issues) for next day or wait for morning standup
  + Make a PR that references correct issues
  + Ensure Master branch has been reviewed by GitMaster on the team, address any comments and deploy to heroku
  + Ensure all pull request have been reviewed by GitMaster
  + Taskmaster ensures no unnecessary tasks are left in progress
  + Group discusses outstanding PRs, review each other’s code and
  + Be sure master is deployable.

Team expectations

* Each of us reserves the rights to our nights and weekends, and we have all  will discuss if it’s needed on friday
* Evening/weekend work policy: at EOD meetup we may volunteer to finish or debug a feature overnight that we were working on, if we feel that will make the next day more productive. No one will begin whole new features or jump ahead to the next stage of the project outside of the work day hours, unless it becomes clear later on that that is necessary to finish on time. Tasks to be done over a weekend will be discussed and planned out on the last weekday. All team members are able to take most of one entire day off every weekend, at least.
* If you have a blocking issue, slack before interrupting another’s train of coding-thought
* None of us will specialize in terms of front-end and back-end code
* Don’t interrupt - raise your hand
* Share the mic
* Disagree often and discuss -- argue infrequently-- Keep debates friendly
* At least message the group when you change what you are working on

Issues

* If the group is divided on a certain decision we will take the following steps:
  + Take 10-15 minute break
  + List out the pros / cons
  + Vote on it
* 4. If the vote is split, escalate to a fellow or instructor
  + If you are frustrated
  + Take a walk or get some water
  + Talk constructively about actionable items (e.g. raise your hand instead of
  + interrupting, let me learn and struggle until I ask you for help, etc.)
  + Bring in a 3rd party (e.g. fellow, instructor) to mediate if you feel too frustrated

Establishing Norms

Each group should discuss and provide an answer for the following questions:

* What does this group do when two or more members disagree (?
  + See aforementioned contract rules
* What does this group do when a member is frustrated?
  + See Contract rules above
  + When someone in the group is frustrated
    - Person A- Frustrated Party
      * Actions to take: Calmly voice the issue and concern to person B
      * If needed take a break
      * If not resolved between pair through initial conversation bring to groups attention
    - Person B- Person who is source of  frustration
      * Action to take: Listen calmly and try to see the other’s perspective
      * If needed take a break
      * Explain if there are issues or roadblocks hindering work
* **Teamwork-  make sure as driver or navigator to clearly explain thought processes to ensure mutual understanding of code.**
  + **All Parties should be responsible to ask questions and be sure they are understanding the work being done**
* What does this group do when we merge our work together?
  + See DevOp Meeting notes
* When does this group ask for help?
  + Work for 45 min- hour within pair
    - Strategy: Documentations, Lecture notes, googling, youtube
  + Then seek help of group- 15 mins and implement any new ideas
  + Else seek Fellow (Leslie) help!
* How do we pair?  Taskmaster decision
  + Do members of a pair decide their own norms? Yes
  + Things to discuss in pair  Driver navigator
  + keep a timer
  + switch positions
* How does this group approach work "after hours"
  + As needed
* Is it okay to work on your own after hours?
  + Yes, if necessary or discussed by group
* How do we keep the group informed?
  + Slack

Coding Style Guide

* Pay attention to the linter!
* Use semicolons
* Two spaces -- for indentation
* Trailing commas where possible
* Use const or let over var
* Use require and module.exports in .js files
* Use import and export in .jsx files, unless require makes for cleaner code
* Put import statements at top
* Put the default export at bottom
* Define container components and presentational components in separate files
* Use the ["ducks" pattern](https://github.com/erikras/ducks-modular-redux) for redux
* Name files using lowercase-and-dashes instead of camelCase or components PascalCase, except for when the default export is a class, then use PascalCase
* Define react components as pure functions (instead of classes) whenever possible
* Single quotes for strings – except to avoid escaping
* No unused variables
* Space after keywords if (condition) { ... }
* Space after function name function name (arg) { ... }
* Always use === instead of ==