CodeTwoGo

mobile codewars

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

# Roles

* Project Manager - Oversight
* Taskmaster
* Gitmaster
* Testmaster

# Daily process

* Morning standup started by Project Manager
  + New roles assigned
  + Today’s goals
  + Obstacles forseen
* Pair program on assigned tasks
* Bring up any blocking issues after 15 minutes of spinning your wheels
* At 2:30pm have re-group meeting where you check-in with blockers, status updates and plan for EOD merge
* Project Manager to check in with team members throughout the day on tasks
* End of the day/night debrief – Issues throughout the day and tasks completed

# Team expectations

* Each of us reserves the rights to our nights and weekends, and we have all been straightforward about when and how we can commit to working on the project
* We will assign tasks horizontally
* Try not to specialize in terms of front-end and back-end code

# Issues

* If the group is divided on a certain decision we will take the following steps:

1. Take 10-15 minute break
2. List out the pros / cons
3. Vote on it
4. If the vote is split, escalate to a fellow or instructor

**CONTRIBUTION GUIDE**

The contribution process is...

1. Use waffle.io to manage tasks/issues
2. Make an issue (or multiple issues)
3. Make a PR that references that issue
4. Get it code reviewed by someone on the team, address any comments
5. Merge into dev-master - This will include all comments - Recap during Night Debrief
6. Master will be the clean copy with no dead code included - Confirm with team members what will be merged to final master

Code style guide

* Pay attention to the linter!
* Use semicolons
* Use Tabs
* Trailing commas where possible
* 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
* Consider splitting up any file larger than 100 lines
* Define container components and presentational components in separate files
* Use the "ducks" pattern for redux
* Name files using lowercase-and-dashes instead of camelCase or PascalCase, except for when the default export is a class, then use PascalCase
* Define react components as pure functions (instead of classes) whenever possible