Browser Testing (Functional Testing)

List of Possible Issues

- Do we need to separate the Travis CI into two builds
    - Example will we run into setting Conflicts
- What browsers does Travis support natively
    - how do we install separate versions
        - example maybe some peps out there running old ass versions of firefox
    - How many browsers should we test for and how far back should we go

- How do we push to the current yaml file without interupting our other team members work
- We gota Become Travis CI experts

Research (Selenium)

- https://testingbot.com/support/other/travis-ci
  - Costs money $$$
- https://carlosgabaldon.wordpress.com/articles/running-selenium-based-browser-tests-on-travis-ci/
  - https://docs.travis-ci.com/user/gui-and-headless-browsers/#Using-xvfb-to-Run-Tests-That-Require-a-GUI
- Great Document about testing with a headless server
  - https://docs.travis-ci.com/user/gui-and-headless-browsers/#using-the-chrome-addon-in-the-headless-mode
- Nice Doc Builds with Travis Ci
  - https://docs.travis-ci.com/user/customizing-the-build
- Welp its possible this single post has solved all of our headless issues
  - so from what I've seen so far chrome and firefox have added headless server support and here's a nice write up on how to do it
  - https://www.amihaiemil.com/2017/07/14/selenium-headless-chrome-travis.html
- There's a whole tutorial on how to set up a Selenium WebDriver Test Framework
  - https://octopus.com/blog/selenium/0-toc/webdriver-toc
- Best thing I've found for integrating Selenium
  - https://lkrnac.net/blog/2016/01/run-selenium-tests-on-travisci/

NightWatch

- Nightwatch.js integrates with Travis, free, has good documentation and uses Selenium but not sure if headless
    - http://nightwatchjs.org/
    - Example: https://github.com/zeroasterisk/meteor-travis-ci-nightwatch
        - **(Ricardo)** Looks like in here he's able to install something on the travis CI servers if this is the case and we can think of them as a sort of VM we might be able to build a docker file and load it through the yml and then run whatever we want but then again this gets a little bit tricky. just a thought
        - **(Ricardo)** It also looks like he runs nightOwl from a script the yml file seems to only install whatever meteor is
    - https://medium.freecodecamp.org/how-to-easily-start-automatically-testing-your-website-8629ea8df04a

Research

- Nightwatch uses the WebDriver API to perform the browser automation related tasks, like opening windows and clicking links for instance.
    - WebDriver is a general purpose library for automating web browsers. It was started as part of the Selenium project, which is a popular and comprehensive set of tools for browser automation, initially written for Java but now with support for most programming languages.

- Nightwatch works by communicating over a restful HTTP API with a WebDriver server (such as ChromeDriver or Selenium Server). The protocol is defined by the W3C WebDriver spec, which is derived from JSON Wire protocol
- The Documentation for this is really really good
  - http://nightwatchjs.org/gettingstarted
- Seems like NightWatch uses selenium
  - This is nice since the more I read the more it seems like selenium is the go to for this

Test Cafe

- This looks like everything will be easier but I don't know if we need it
  - https://devexpress.github.io/testcafe/documentation/continuous-integration/travis.html

Research

- This option doesn't use selenium but it appears there are some bonuses to this
    - Unlike most testing solutions, TestCafe is not built on Selenium. This allows us to implement features you cannot find in Selenium-based tools (for example, testing on mobile devices, user roles, automatic waiting, etc.).
    - TestCafe uses a URL-rewriting proxy which allows it to work without the WebDriver. This proxy injects the driver script that emulates user actions into the tested page.
- Seems Super easy to get working in Travis CI

Browser

Chrome Info

- https://docs.travis-ci.com/user/chrome
    - Has info on using chrome in headless mode
- https://stackoverflow.com/questions/40193722/selenium-with-headless-chrome-2016-on-travisci
    - Since Chrome has native headless mode, we might just need to write whats in the answer

Microsoft Edge Info

- https://www.browserstack.com/test-on-microsoft-edge-browser
  - Free testing for edge browser

Firefox Info

- I found a headless mode that's available for Firefox, not sure this helps but I'll put it in just in case
  - https://developer.mozilla.org/en-US/docs/Mozilla/Firefox/Headless_mode

Travis CI

Creating Multiple Test Environments In Travis CI (Should Solve our Multiple Browser Testing Issue)

- Below code was found on https://gist.github.com/jamesarosen/e29076bd81a099f0f72e this piece of information is also reflected on the official docs @ https://docs.travis-ci.com/user/customizing-the-build you can see they recommend using these things called build marixs

```
matrix: include: - php: 5.6 env: TO_TEST=FRONTEND - php: 5.6 env:
TO_TEST=BACKEND fast_finish: true install: - if [ "$TO_TEST" =
"BACKEND" ]; then some_backend_command; fi - if [ "$TO_TEST" =
"FRONTEND" ]; then some_frontend_command; fi script: - if [ "$TO_TEST"
= "BACKEND" ]; then some_backend_command; fi - if [ "$TO_TEST" =
"FRONTEND" ]; then some_frontend_command; fi - if [ "$TO_TEST" =
"FRONTEND" ]; then some_other_frontend_command; fi
```

Plain Text