Cushion Testing Assignment

Assumptions

- API is implemented in PHP
- API behaves as drafted in Activity Diagram
- API receives changes through a configured CI pipeline for releasing
- API path to update monthly contribution accepts “amount” or “percentage” in body

Proposal

The API that enables customers to change their monthly contributions requires automation coverage. The value in implementing an automated solution means the team can get fast feedback when improvements are made to the API and it means time can be spent on other functionality that requires manual testing.
Automated tests should be implemented as followed:

- Unit tests: Tests should be designed, implemented and seen failing for specific units of code in PHP using PHPUnit
- Integration tests: Tests should be designed and implemented calling the API in expected request formats and response should return expected data. Implementing Newman will enable the postman collection to run headless and on CI. Refer to postman collection of what is expected
- End to End tests: Tests should be designed and implemented in a way that imitates how the end user will interact with the API from the view, this can be achieved with Selenium or Appium

Enhancements

- Pre request script that is able to generate dynamic user when it is required
- Unhappy test cases should be implemented to ensure the correct error code and error messages are in response
- Automated tests should be able to CI to be able to execute as and when there are changes to the API
- Create environment data sets to manage the different data sets between each environment
- Use request chaining to extract required variables to be used in different requests
- Performance testing the API to see how the API handles different loads
- Does the API need to send any additional data for Accessibility testing
- Any other scenarios that are currently being manual tested it would be good to automate