StandUp 1 - HashGraph Project

https://github.com/olliegilbey/HashgraphVisualiser

1 User Stories

1.1 User Story 1

As a user, I want a web based interface, so that I can view it in a browser.

- Flask webserver
- HTML page
- Python integration of HTML and Flask
- Must display in Firefox and Chrome
- Server must be stable

1.2 User Story 2

As a user, I want the graph to dynamically update, so that I don't have to constantly refresh the page.

- Only the graph must update on the HTML page
- The graph must only update when a new transaction occurs

1.3 User Story 3

As a user, I want to be able to change the number of nodes visualised, so that the width of gossip can be visualised at varying scale.

- Slider bar on the web interface to change number of nodes.
- Implement functionality into data extraction.
- Test to find max number of nodes for processing.
- Limit slider to the max number of nodes.

1.4 User Story 4

As a user, I want a start and stop button, so that I can capture the graph over a specific period of transactions.

- On click of start button, restart the live data extraction.
- On click of start button, the visual must be restarted.
- On click of stop button, the visual must stop.

1.5 User Story 5

As a user, I want a save button to export the graph findings, so that I can save the resulting session for comparison later.

- On save button click, export the visual to an image file.

1.6 User Story 6

As a user, I want a visual representation of the HashGraph, so that I can verify finality.

- The graph data must be available.
- The graph data must undergo a conversion for the visual.
- Graph theory must be applied.
- Graph must be plotted.
- Graph must be displayed in web interface.