Sprint 1 Team 9 Hashgraph Backlog

Wernich Baumgarten <19896913 >
Simon Dennis <19770235 >
Oliver Gilbey 17839688 >
Ryan Lang <19066171 >
Simoné van Zyl <18395317 >
Bernard Wiesner <19790155 >

1 Sprint 1 Backlog

1.1 US1: As a user, I want a web based interface, so that I can view it in a browser.

Completed

- Flask webserver
- HTML page
- Python integration of HTML and Flask
- Must display in Firefox and Chrome

Incomplete

- Server must be stable

1.2 US2: As a user, I want the graph to dynamically update, so that I don't have to constantly refresh the page.

Completed

- The graph must only update when a new transaction occurs
- Graph requires X, Y co-ordinates

Incomplete

- Only the graph must update on the HTML page

1.3 US3: 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.

Completed

- Slider bar on the web interface to change the number of nodes

Incomplete

- Implement functionality into data extraction.
- Test to find max number of nodes for processing.
- Limit slider to the max number of nodes.

1.4 US4: As a user, I want a start and stop button, so that I can capture the graph over a specific period of transactions.

Completed
None
Incomplete

- 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 US5: As a user, I want a save button to export the graph findings, so that I can save the resulting session for comparison later.

Completed
None
Incomplete

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

1.6 US6: As a user, I want a visual representation of the HashGraph, so that I can verify finality.

Completed

- The graph data must be available
- Graph must be plotted.
- Graph must be displayed in web interface.

Incomplete

- Graph theory must be applied.