

**COP5615: Distributed Operating Systems, Fall 2018**

Project Report

Bitcoin Simulator (Part 2)

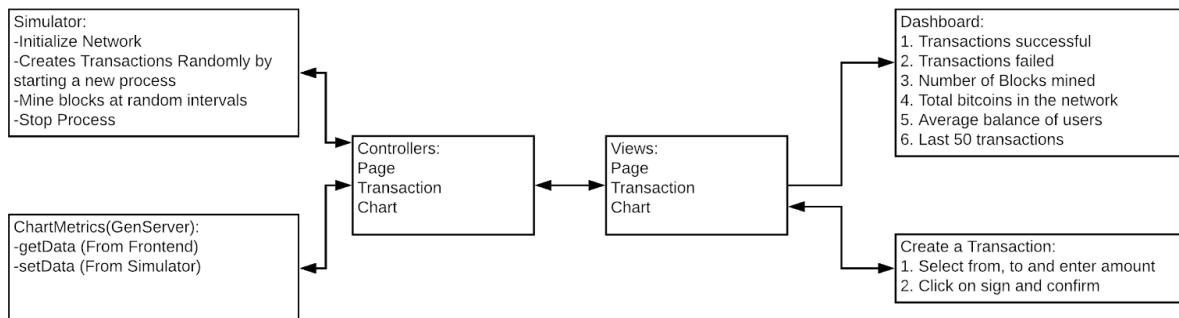
Submitted by:

Kunwardeep Singh (UFID: 2421-3955)

Gayatri Behera (UFID: 3258-9909)

# 1. Description

For this project, we have used the Phoenix framework to create a web application using a dashboard interface that monitors the bitcoin network in the backend. Phoenix's MVC architecture is used to design layouts, web pages and the controllers necessary to communicate with the back-end and relay the information back to the front-end.



## Backend:

The same functionality of 4.1 project is being used here. Apart from that, a simulator module and a metrics module is implemented. The simulator module initializes the number of nodes and creates random transactions at random intervals. The chart metrics module keeps track of transactions, amounts and the blocks mined. All metrics are exposed by a single method `get_data()` that sends the current data collected by chart metrics module to the client.

## Frontend:

An input field is created for entering the number of nodes in the network along with the buttons to start and stop. The following charts were displayed:

1. Number of successful transactions
2. Number of failed transactions
3. Number of blocks mined
4. Total bitcoins in network
5. Average wallet balance per user

The charts are implemented using `chart.js` that updates the current metrics data using the `get_data()` method. Along with these charts, last 50 transaction ids with amount are also being updated after every 5s.

### 3. Bonus Features

In this part, a page is created for creating a transaction along with signature by the sender.

The flow is as follows:

- Select sender and receiver public keys
- Upon selecting sender public key, sender's wallet balance is displayed.
- Enter chosen transaction amount
- Sign and Confirm transaction
- View Signature, generated transaction ID and status of confirmed transaction
- The transaction ID can also be seen in the Dashboard