Vault

A SHARED DISTRIBUTED AND REDUNDANT STORAGE SOLUTION

Progress so far...

Group

TRNR Peiris - IT 16106420

KVA Sachintha - IT 16158528

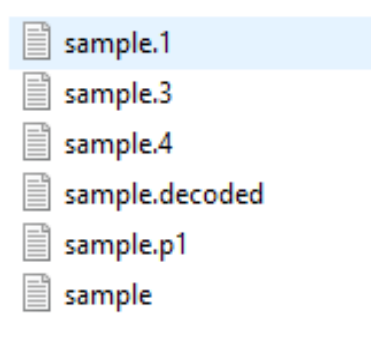
BA Ganegoda

WMUKMT Bandara-IT16091276

Node Authentication, encryption and File Sharing module (BA Ganegoda)

- Encryption and decryption functions in progress(symmetric encryption) (50%)
- Sharing part was broken down into two parts as Public and private sharing and not yet started
- User authentication in progress(50%)
- Node authentication and integrity validation not yet started.
- Content based search algorithm for faster file search is in mind.

File Redundancy and Disk Health Monitoring (TRNR Peiris)



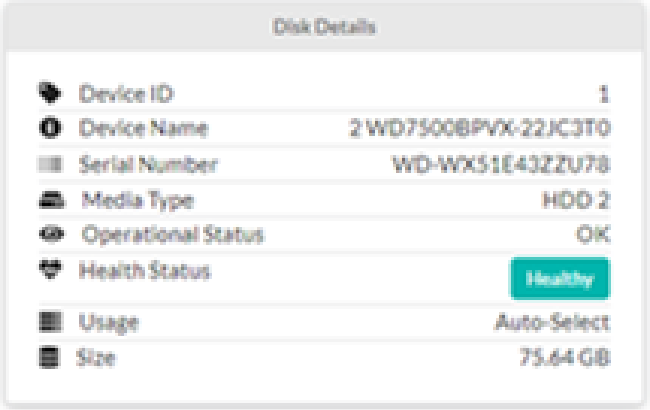
- One file scattered into 4 data shards - Completed
- Generating parity shards - Completed
- Regenerating file from any 4 out of 6 shards available at the time - Completed
- Reducing the amount of space required for a parity shard - In progress
  - Size of a parity shard = size of a shard + size of a shard /8 Completed
  - Size of a parity shard = size of a shard - In Progress
- Performance optimization - In progress

Disk Health Identification

Modern hard drives have “S.M.A.R.T..” It's a feature that allows operating systems

(like Linux, Mac, and Windows)

to verify the integrity and health of the hard drives.



- Widows - Tested
- Linux - Tested
- MacOS - Scheduled
- Frontend - Completed

Blockchain based DHT and Messaging Protocol (KVA Sachintha)

```
let Block = thinky.createModel("blockchain", {
    id: type.string(),
    index: type.number().integer(),
    previousHash: type.string(),
    hash: type.string(), // hash of block, timestamp and prev hash
    timestamp: type.date(),
    block: {
        filename: type.string(),
        filehash: type.string(),
        filesize: type.string(),
        fragment_count: type.number().integer(),
    },
    metadata: {
        owner: type.string(),
        deleted: type.boolean(),
        enc_key: type.string(),
    }
});
Block.ensureIndex('index');
```

- Blockchain core logic complete.
- Block Verification complete.
- Functions for calculating Merkle root in progress.
- Blocks are currently persisted to a NoSQL database.
- Advantages of using an RDBMS is still in mind.
- Currently designing the API that is eventually going to be exposed

Messenger Module

client.js

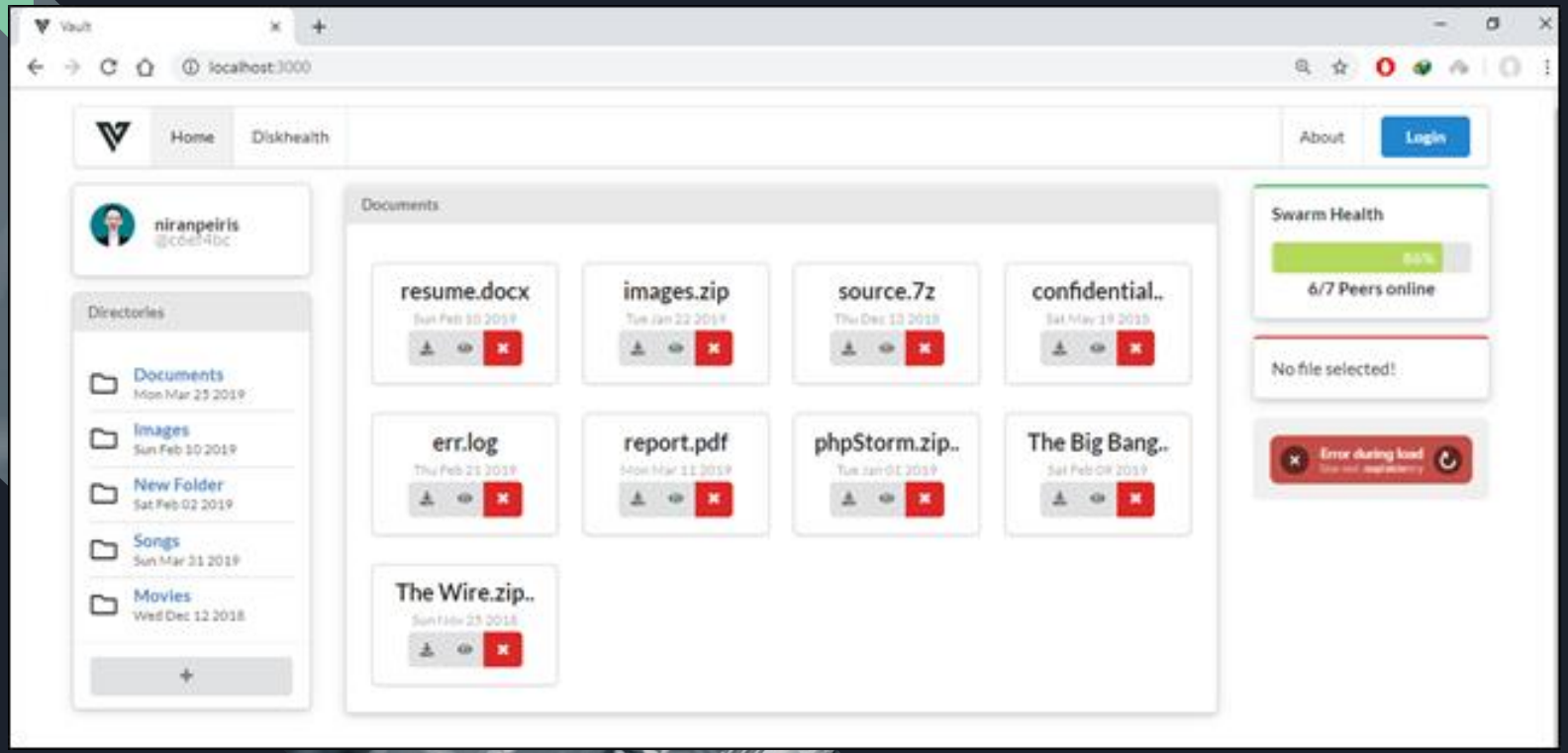
```
"use strict";
const io = require('socket.io-client');
const portscanner = require('portscanner');

let ipArr = [];
let sockArr = [];
const protocol = 'http://';
const port = ':3000';


let addSocket = function (ip) {
    ipArr.push(ip);
    let sockAddr = protocol + ipArr[ipArr.length - 1] + port;
    let socket = io.connect(sockAddr);
    sockArr.push(socket);
};


module.exports = {
    addSocket,
    sockArr
};
```

- Uses Socket.io transport for websocket based 2 way communication
- Uses 2 channels
    - For Blockchain updates
    - For Configuration updates
- Currently testing automatic detection of messenger service via portscan.
- Currently formulating a Consistent Messenger API for external modules to use.
- Currently testing various heartbeat intervals vs performance degradations.

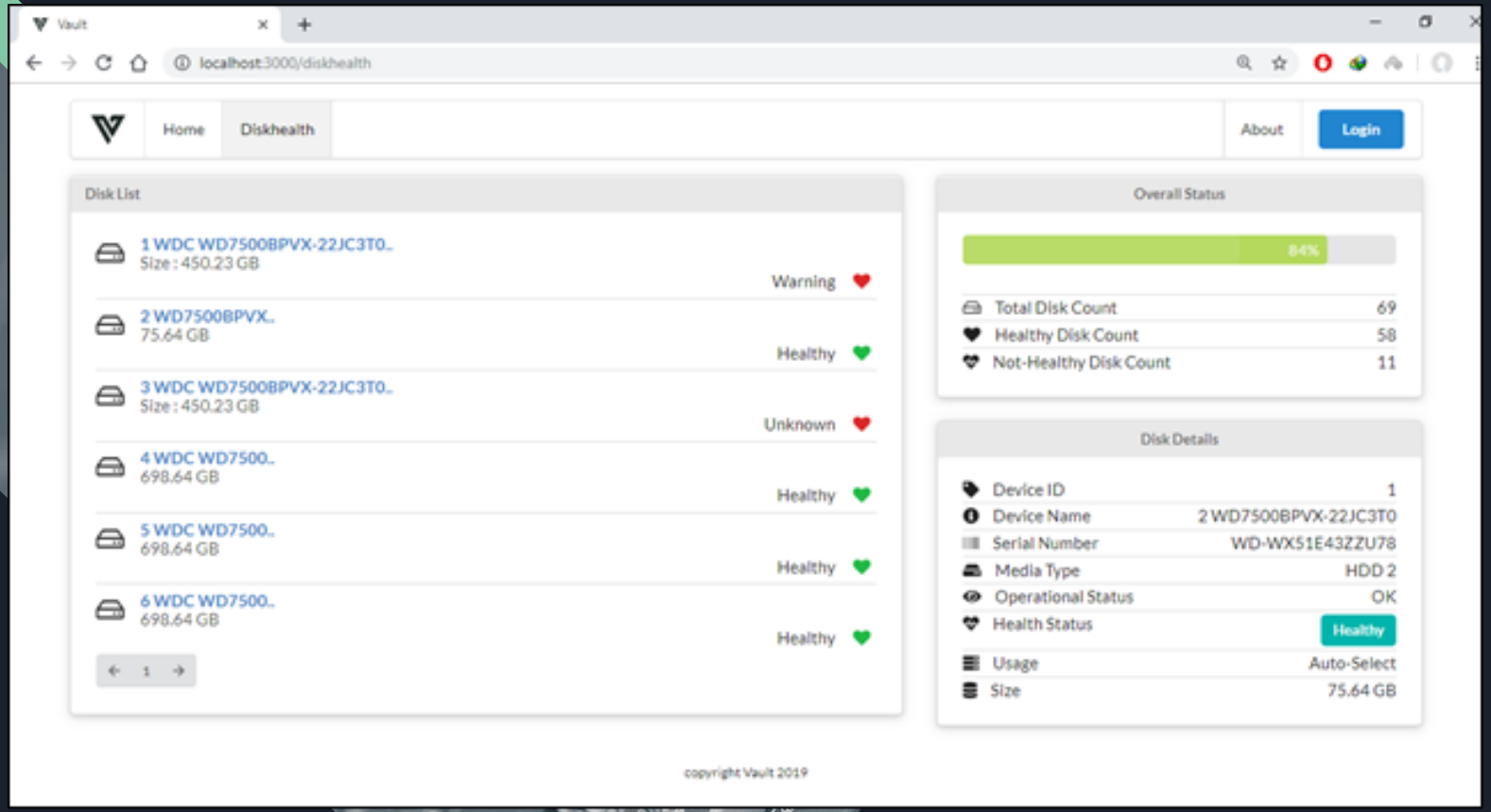
Static and Dynamic Disk Space Allocation methodology (WMUKMT Bandara)

- Virtual hard disk creations.
  - create virtual disk vmalloc() function. (in progress)
  - Location initialized to 0 memset(). (in progress)
- Develop a device driver for virtual hard disk. (not yet implement)
- Device files operations
  - Read / write is implemented using memcpy() API. (Not yet implement)
- Block operation. (in progress)
  - The request function to process the requests in the request queue.
  - The spin lock associated with the request queue to protect concurrent access.
- Disk status monitoring. (front end complete)

Main UI - Normal Users Interface



Main UI - Disk Health



Thank you!