Activity A – Service Status Monitoring Solution

**Proposal for the digital solution:**

To meet the requirements of the digital solution, I will be creating a web based dash board using python and flask that will interface and query the college’s existing systems to check the status and availability of each of the systems, this will then be securely fed back to the developed software and displayed using a traffic light system which can be manually changed by authorised figures in case the query returns false information because the software says it is up but people can’t access it. This will allow any one with right access to check the status of all machines and servers without manually checking on all of them, this allows the IT team to find and diagnose problems much faster, infinitely decreasing downtime.

**Function requirements:**

* Web based
* Traffic light system
* Record uptime
* Get status and query existing
* Display previews to user

**Non-Functional requirements:**

* Large Icons
* Utilizing information hierarchy
* Constant spacing of icons and widgets
* Reasonable font and font size

**Problem Decomposition**

**Web based**

Must use Python with Flask to produce a dynamic website that will pull data from existing systems and integrate Python into HTML.

**Security**

Will use Flasks built in forms module to build and implement a log in screen so that only those that are granted access by admins will be able to access, see the dashboard and edit the statuses of the servers.

**Querying and accessing the servers**

Will use SQLite to query

the servers to retrieve their statuses and information before using this data to generate the stoplight (RAB) to indicate its status.

**KPI’s**

* Main modules developed (login, register, server pinging, querying ticket system) should take a week each.
* Small modules completed (converting server feedback to traffic light, wireframes for each page) should take a couple days each, wireframes less than a day per frame.
* Percent of users happy with product vs users who want more work/features.

**User Acceptance criteria**

* Must have all the functional requirements within the software
* Must be easy to navigate so no training is needed for the software
* Must have a colour theme that matches the colleges colour scheme and works well
* Must allow user with access to submit feedback on server or websites that are down in case they are showing as up when it is pinged

**How does the solution meet the Colleges need?**

The college needs a way of making it obvious when a server or website fails and goes down, this web-based service will allow people to make edits when a server or website stops working as well as pinging them itself to dynamically change the server’s status on the website when a server experiences downtime.

**Risks**

One of the biggest risks when working with existing systems and college servers is interpting any tasks that thye are performing my interfering so to get around this we will only work with one system at a time so should anything go wrong we will know what caused it and how to reverse it.