Analysis Page View (page details):

1. The base route should be based of an Azure Services (VM, Storage, etc: /analysis/vm/<resourceid>). This is because each resource requires a different set of Questionnaires to perform an analysis. We should have a component for each of these cases (folder: analysis/vm.js, analysis/storage.js etc..). Analysis is essentially a set of questionnaires that a consultant should be able to response.
2. Each Question should be provided a chart or an image (for now lets just do chart) and a “comments” section. For the VM requirements these are the questions:

Tabs for the page details

**Compute**

* 1. What is the CPU Utilization Analysis? (Compute)
  2. What is the Memory Analysis? (Compute)

**Storage**

* 1. What is the Disk IOPs and Latency analysis? (Storage)

**Network Data**

* 1. What is the Bytes Send/Received Rate analysis? (Network Data)

**Migration and Agreements**

* 1. Can you reserve this instance of VM? (cost benefits)
  2. Can you work with Hybrid benefits?
     1. Enable existing on-premises licenses with software assurance to be leverage in azure.
     2. Windows Server
        1. Standard – Each 2 processor / 16 core license allows two 8 core or 1 16 core instance
        2. Datacenter – as above but allows simultaneous use on-premises as well
  3. Any concerns with SLA (Service Level Agreements)?
  4. Are there any Migration pending for this VM?
     1. Propagating data between on-premise and Cloud can have performance issues. Best is always to migrate all it in the cloud (even if you have a great spress-routes).

**Optimization**

* 1. Can you optimize any service as PaaS services (PaaS < IaaS)?
  2. Do you have any scale up/down rules in place? (Vertically scaling)
     1. Pro: increase scale you are more powerful
     2. Cons:
        1. requires downtime to resize
        2. No resiliency
  3. \*\*Scaling Out (Horizontally- dynamic scale). This typically the desire state
     1. Pro:
        1. Increased scale,
        2. Increase resiliency
        3. No downtime required
     2. Con:
        1. Load balancer (new cost)
        2. App may have to be changed to work multi-instance

**Accountability and Governance**

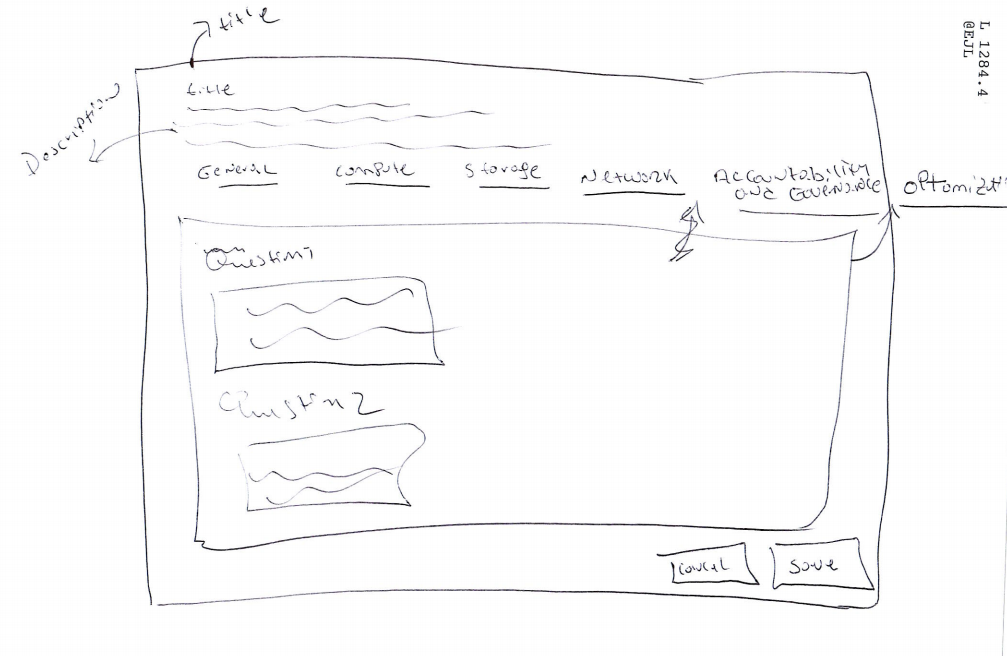
* + 1. Project name?
    2. Cost Center?
    3. Who is the creator?
    4. Who is the owner?
    5. Do we have any Azure policy to prohibit expensive Resources?
    - Organizations Struggled with VM Sprawl and were unable to identify owners of resources which often were no longer needed.
    - It is critical to ensure all resources can be tracked to a project, owner and commonly cost center
    - A logical structure around management groups, Subscriptions and Resource groups will help with accountability and governance
    - Tagging enables per resource metadata that can be required and inherited from the parent resource group via Azure policy (cost center, creator, etc). Azure policy can help you to re-inforce these rules.

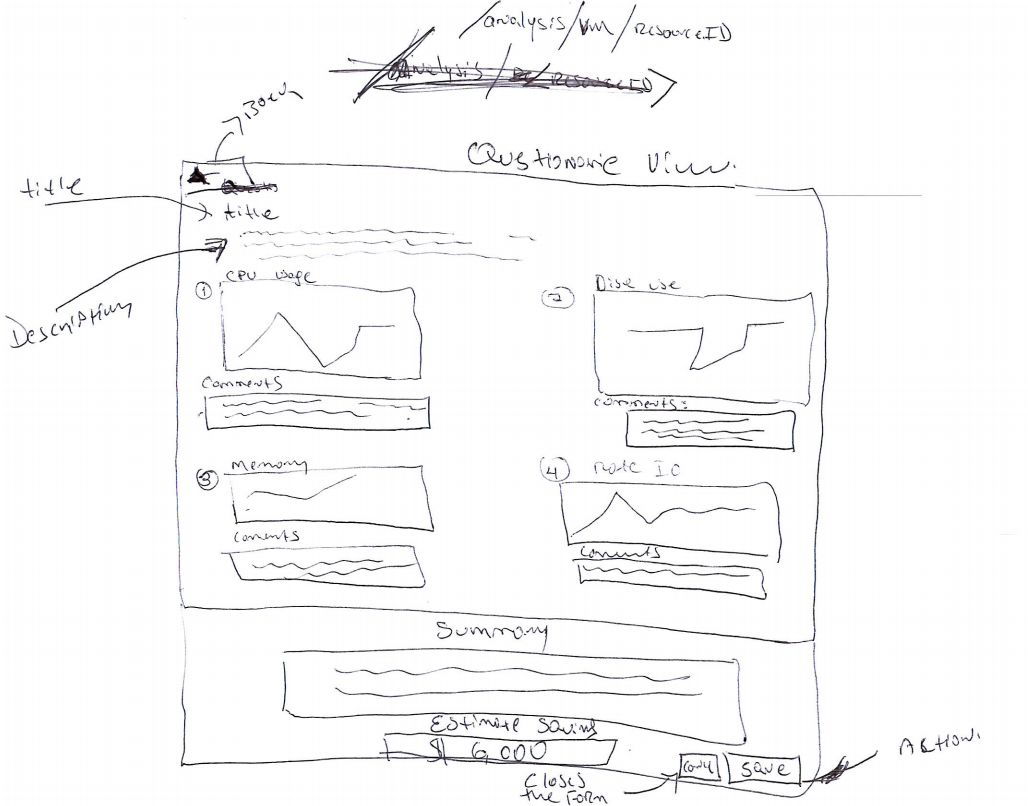
**Evaluation/Summary**:

Summary: the sku be changed. Blabla

Estimated Cost Saving: $800

* Try to get as far as SaaS as possible which minimizes customer responsibility and therefore cost
* Compute
  + VM -> VMSS-> Containers -> AKS with Containers -> App Service Plans -> Serveless
* Database
  + Azure SQL -> Cosmos DB -> IaaS VM
* For all workloads understand the usage pattern including Peaks and duration.





Graph example:

