Project Overview:

- Glossary:
  - Performance - Representation of the degree of optimization.
    - optimization of what? Memory usage, i/o, cpu cycles, avg retrieval time of data in DB, avg insert time into DB, YCSB score(s), weighted average of multiple?

Requirements Specification:

- Use Cases:
  - only one use case in this document that is the entire project, "Use Case 1"
  - Split into multiple use cases, not one monolithic use case. For example:
    - View a report of the results? (Sprint 2 goal for Xuan)
    - "One touch" deployment or some user specified options (how optimized for example)?
  - Exception path is described as N/A, will it never fail, Handle failure gracefully, or crash and leave the VM's in an unknown state?
  - Input is N/A, is the docker image (containing Cassandra) mentioned in the brief description obtained by the program/script or by the user?
  - Output is N/A, is the output not an optimized Cassandra DB running in a Docker container in this use case?
- User Stories:
  - None documented, should have some to provide context for the use cases.
  - Should be follow the I.N.V.E.S.T. acronym (Independent, Negotiable, Valuable, Estimatable, Small, Testable)
  - Example:
    - As a database administrator I would like to one click deploy a cluster
    - As a database administrator I would like a database deployment to optimize for <optimization criteria, see glossary suggestion for examples>>
    - As a system administrator I would like the deployment to be contained in Docker containers

High-Level Design:

- UML Diagram:
  - No Legend
  - Data is shown to flow one way from the Parameter Generator to the deployer. In the spec it says that the deployer will (eventually) optimize the parameters but no data is passed back to the parameter generator to update/optimize the values. Does it randomly generate? Document this somewhere.
  - Data is shown to flow from the report to the visualizer to the benchmarking tool. Should this not be the other way around?

- Text under diagram:
    - Parameter Generator is described that it might get data from the visualizer, the Monte Carlo method requires that results are returned to it so that it can move in the right direction. This is also not reflected in the diagram.
    - YCSB is described as running on each of the VM's, the diagram depicts it running on the master machine.
- Storyboard / UI
    - Not necessary due to the nature of the project.
    - Analysis Report is the only visual aspect of the project. It is laid out logically and information is presented in an easy to read manner.

Release Planning:

- Gannt charts are supposed to be broken down into individual tasks under a larger heading (Sprints, Components, or Categories for example) then assigned to people, your Gannt chart is too general.
- See https://en.wikipedia.org/wiki/Gantt_chart for more information.
- Ensure to mention predecessors/dependants when breaking the Gannt chart into tasks, as well as highlighting the critical path.
- Dedicated tools such as Microsoft Project, or Spreadsheet tools such as Microsoft Excel or Google Sheets are great tools for developing Gaant Charts.