General Notes:

- There does not seem to be a master installation or usage guide anywhere, it is probably a good idea to create an installation guide as well as a usage guide.
- The repository may need to be cleaned up a bit, for example in the parser there is an all.sh, as well as a directory called all with an all.sh and an allv2.sh

Requirements:

- Diagrams have been updated nicely as per earlier suggestions
- Gantt chart much better
    - (minor) could be broken down into more specific tasks, it is still pretty generalized at the moment
- (minor) On the Requirements specification page there is still only one use case, either a link to the User Stories page and the Use Case page should be provided, or the use cases and user stories should be combined / moved.
- Glossary has been clarified as suggested.
- New addition of UI mock-ups and the storyboard / navigation diagram accurately displays a functional UI
    - Display's the contents of all screens nicely as well as transitions between the screens.
- Use Cases:
    - (Minor) grammar could be improved
    - (Intermediate) no exception paths provided in any use cases.
        - Example exception path for UC-5: unable to connect to database (database is down or off or there is a network issue if the database is on a different machine)
- User stories:
    - (Minor) grammar could be improved

Code Base:

- Deployer:
    - README.md
        - Description is well defined and accurately describes what the script is for.
        - All input parameters and options are covered in the usage section.
    - smartDeployer.sh
        - code style is consistent throughout the script and with script guidelines
        - sufficient comments for major code features allowing easy navigation for the reader.
        - As of now the deployer does not seem to be connected to the optimization system as both options for -o (ON or OFF) do the same thing, but this is to be expected as it is not scheduled until sprint 3.
- Parameter Generator:
    - The goal of this part of the project for Sprint 2 was to have an idea of how to accomplish this task, get a finalization or the requirements from the client, and start writing a template for the functionality.
    - README.md

    - Documentation clearly indicates that investigation has been made into what settings can be configured for both Docker and Cassandra to optimize performance.
    - Example output in the form of an image, and sample input in the form of a text file have been provided.
  - 401test.py
    - It is very clear that this file is only a test file and a potential template for the development in sprint 3.
    - Variable naming conventions are consistent throughout the file, as well as clear as to what the variable is for.
    - (minor) Spacing of brackets around functions could be more consistent
- Parser:
  - The parser according to the high level design is one module / program, but there are many parts in both shell scripts as well as python scripts. Some extra design documentation should be created to reflect the actual structure of the module.
  - README.md
    - Does not make it clear if the script is to be run separately or on its own, or if it automatically invoked as part of the deployment process.
    - There is a note about not needing to run "install ycsb.sh" if its already installed, does it get installed automatically, or is it a manual installation required or if is installed by another script somewhere.
  - In_container_ycsb.sh
    - code style is consistent throughout the script and with script guidelines
    - sufficient comments for major code features allowing easy navigation for the reader.
    - (minor) comments have some grammatical errors and awkward structure.
  - all.sh
    - This appears to be an old version of the all.sh file from the all directory? Review for it will not be completed.
  - getHostIP.py
    - At the moment this program only reads a file and returns a value from it.
  - install ycsb.sh
    - Simple script to install ycsb
    - Well commented the block of comments at the top appear to be for a different file though, there are references to required files that are not referenced anywhere in the script, as well as indication of data being generated but there is no such data generated in the file.
  - generate_report.py
    - i is assigned values that appear to contradict with each other in lines 13 and 25.
    - Line 13 should be for i in fileOrder to maintain consistency if the file order changes.
    - As of now this file just appears to spit out text, the reference to matplotlib.pyplot is not used anywhere.
    - Unfinished? Or not being used anymore?

        - The only references to this file are in the all.sh files and the allv2.sh file where the same data is also sent to the database.
    - All Folder
        - This folder appears to be an expanded and/or completed version of the all.sh file above
        - It appears that the parser runs YCSB, the infrastructure documents should be updated to indicate this as opposed to just showing ycsb as a framework.
        - README.md
            - Usage is documented, but default, example or expected values are not given.
            - It is not clear how step 2.1 fits into the scheme of things.
        - UnitTest.py
            - (intermediate) Root username and password stored in cleartext in the python file, it would be more secure to create a new user with only the required permissions for inserting / updating to specified containers. This is somewhat mitigated since the database is run on a “secure” server, but this practice should be avoided in general.
            -
        - All.sh
            - Mysql connector is installed (or attempts to install) every time all.sh is run? Is there a reason for this?
            - Appears to be an old version or soon to be depreciated version with allv2 being the successor?
        - Allv2.sh
            - ycsb is downloaded and attempts to be installed every time this script is run? Is this script ment to be run only once, or will it be run multiple times?
            - Will there ever be a use case to use this on a network without an internet connection?
            - Script is well commented and conforms to style found in other scripts in the project.
        - Commit_to_database.py
            - (intermediate) Root username and password stored in cleartext in the python file, it would be more secure to create a new user with only the required permissions for inserting / updating to specified containers. This is somewhat mitigated since the database is run on a “secure” server, but this practice should be avoided in general.
            - Code is structured nicely in a logical manner
            -
        - Generate_report.py
            - This file is identical to the generate_report.py file in the directory above, with the exception of an extra newline on line 8.
- Resources:
    - README.md

  - SmartDeployer.sh
    - This appears to be another version of the Deployer/smartDeployer.sh file? Which one is the actual script? Combine the two, or provide an documentation to indicate which one is for where.
    - This one appears to be more fleshed out on the optimization side, but doesn't have as many options.
    - Gets stuck on running "sudo docker run Cassandra", after manually stopping Cassandra there is a syntax error further in the file.
  - 401project.sh
    - Running this script does not complete, the progress stops at running Cassandra. After quitting Cassandra the script completes as expected.
  - In Container.sh
    - Is meant to be run inside the container, no explanation on how to get it into the container is provided.
  - In VM.sh
    - Does not run, syntax error.
- Web:
  - By convention including a requirements.txt file should be supplied to ease installation of the Django environment, one can be generated with "pip freeze > requirements.txt" or "pip3 freeze > requirements.txt" on python3. The environment can then easily be installed with "pip install -r requirements.txt" or "pip3 install -r requirements.txt"
  - Notes:
    - Style should be implemented in a single, or multiple CSS files then referenced by the html files to maintain visual consistency across the entire website, as well as to avoid code duplication. The implementation of style in CSS files also closer conforms to standard web development practice.
    - There appears to be file duplication between /web/smartDeployer/results/templates/ and web/frount end/
  - Front End:
    - There is a typo in the directory name, Web/Frount end -> Web/Front End
    - Templates for the about, help, and team pages have been created.
    - Template for the real results page has been created, but all scripts are included in the html file, it would be more extensible to have a separate script file with the javascript in it and allow the html file to focus entirely on layout (separation of concerns)
    - If the graphs are not meant to be interactive another potential way to accomplish the graphing would be to generate the graph In python, then send it to the front end. An argument can be made for generating the graphs in the front end or the back end though, for this project it is simply a design decision.
  - Smart Deployer:
    - It appears that for now the entirety of the smartDeployer Django app's only functionality is to display the sample_results.html template. In the Release planning all's that was scheduled to be completed was setting up the initial UI for the web application. Should the links along the bottom of the page be setup

to navigate to the already created about, help, and team pages? There are no entries in the urls.py for them.

Testing:

- There is only one formal test provided and it is duplicated in two places (Parser/all/UnitTest.py and Tests/UnitTest.py). It only connects to the database, selects data from all of the tables, and prints it all.
- More formal test cases should be developed when more of the website has been created.
- smartDeployer.sh
  - correctly validates input for known flags
  - unexpected flags cause the program to hang as it does not shift past them ("./smartDeployer -f" for example)
- allv2.sh
  - unexpected flags cause the program to hang as it does not shift past them
  - there is an index out of range exception in the commit_to_database.py invoked by this script.