Requirements

|  |  |
| --- | --- |
| Name | Task |
| Francois van der Merwe | Integration between the Execution Unit and Interface Unit.  This requires:   * Ingesting tasks from a queue on the Execution Unit’s team. * Responding to Execution unit that the task was successfully ingested * Updating data in Node Information such as the node status. Insert new record into the Task Table. Updating records such as The node that is working on the task, The Metric Type and Start Time * Calling the task manager * Fetching the result and updating the task table with the metric value and the end time * Putting it on a queue for the Execution Unit to receive * After successful feedback from Execution Unit. Return True Boolean for node to status to be updated by task manager |
| Kathutshelo | Creating the task manager by mocking out the task execution process  This Requires:   * Receiving a task from the node handler * Mocking the execution of the task by returning a value for each of the difference metrics * Populating a result class with the single metric * Passing the result to the node handler * Check the node calendar and update node status accordingly. (If node is not supposed to be working set node status to unavailable status otherwise set node to ready status) |
| Rebecca | Creating the node manager and communicating with the database  This Requires:   * MVC pattern where the view is a simple interface which will be given to the Interface Unit. * The controller handles the connection to the database. * The interface will include 4 tasks * Approve node, updates the Node Information Table and Node Calendar Table, starts a new thread which acts as a node. * Kill node, which would stop the node with its current task, updates the task table by setting the End Time to the current time and set the node status to unavailable in the Node information table. * View node, fetches all nodes and its statuses from the database * Remove node, stops a thread and removes the nodes information from all three tables. |
| Martha | Will implement database and all its tables.  This requires:   * Three tables: Node Information, Node Calendar, Task Information * Node information has 3 comlunds.  1. Node ID as the Primary Key 2. Node Creation Date 3. Node Status  * Node Calendar has 4 columns  1. ID as the Primary Key 2. Node ID as the Foreign Key 3. Node Active Start Time 4. Node Active Stop Time  * Task Information has 6 columns  1. Task ID as the Primary Key 2. Node ID as the Foreign Key 3. Metric Type 4. Metric Value 5. Start Time 6. End Time  * Relationships between tables need to be created such as one-to-many, many-to-many or one-to-one * Each primary key needs to be generated uniquely. This is achieved by sequence |