**File Management Class:**

Handles reading from file and writing to file.

read\_input\_files() {

}

write\_output\_sucess() {

}

write\_output\_fail(failure\_step) {

Input parameter is what stage failed.

Method writes “Failure at failure\_step”

}

**Map Class:**

The map class will have three functions: map, map\_intermediate\_export, and map\_export.   
 The map function will take in two parameters,key and val ue. The map function will tokenize the value parameter into distinct words. Then, it will put it in the format of (“the”, 1), (“a”, 1), etc…   
 The map\_intermediate\_export will take in one parameter, map\_input\_parameter. The map\_intermediate\_export function is responsible for writing input to the intermediate file.

The map\_export function will take in two parameters, key and value. This function is responsible for buffering output in memory and to periodically write to disk.

**Sort :**

sort() {

Reads data from the intermediate file

Combines all input into combined pairs.   
 (the, [1, 1, 1] ), (“a”, [1, 1, 1 ])

}  
  
**Reduce:**

Reduce () {

Takes input from sort and then reduces

}

reduce\_export(key, resultingReducedValue) {

Writes result to output directory.

}

**Workflow Component:**

The workflow component is what the program will use to determine the order of calls. The workflow component is responsible for keeping a synchronous chain of events. Each method called by the workflow component will return a value to determine success, which will allow the workflow component to call the next method in the chain. The workflow component will call seven functions in order. The first six calls are predetermined and a conditional will determine the call for the seventh function. The first six functions in order are:   
 read\_input\_files

initiate\_map  
 initiate\_map\_intermediate\_export

initiate\_map\_export

initiate\_sort

initiate\_reduce  
 initiate\_reduce\_export

If the result of initiate\_reduce\_export is successful, then the workflow component will call write\_output\_success. If unsuccessful, the workflow component will call write\_output\_fail.

**Calls File Management Class**

read\_input\_files() {}

**Calls Map**

initiate\_map() { }

**Calls map\_intermediate\_export**

initiate\_map\_intermediate\_export() {}

**Calls map\_export**

initiate\_map\_export() {}

**Calls Sort**

initiate\_sort() {}  
**Calls Reduce**

initiate\_reduce() {}

**Calls reduce\_export**

initiate\_reduce\_export() {}

**If reduce\_export is successful, call write\_output\_sucess**

**Else, write\_output\_fail** success\_check() {

If (success) {

write\_output\_success()

} else {

write\_output\_fail()

}

}

Executive Component:

Passes command line arguments  
 Calls Workflow component