9. FAQ

1. **Can we give an unsorted, randomly ordered arraylist in newuser and newproject or should we sort them like we sorted newprojectuser?**

Ans: Only projectuser is asked to be sorted in the specs.

Importing maps are allowed ?
Ans: NO! You can import java.util.ArrayList, java.util.Stack, and java.util.Queue. You are also allowed to import list, and linked list! (08/10/2019)

How the correctness of flush query be checked as it does not return anything ?
Ans: The final state of the system will decide whether it had executed correctly, as the completed jobs are printed in the order jobs were executed.

What do we have to return in timed_user etc. ?
Ans: Return an empty list.

What is the memory limit of eclipse? How much space are we allowed to use ?
Ans: Don't worry about this, we will ensure that it's executing for majority of the class.

Do we need to add ID of Jobs as given in the output file in mail?
Ans: It's no required, it's totally optional.

Is there any change in input/output file?
Ans: yes! you can donload it from link just below , Please refer the "Flush specifications" piazza post regarding further clearification on new input/output.
InputOutputNew.zip (14/10/2019)

Assignment 5
Project Scheduler Queries

Release date: **4th October**
Max marks:
Due date: **16th October**
Submissions on Moodle.

1. **Brief description**

This assignment is an extension to the Project Management (Scheduler) part from **assignment 4**. You are required to perform more complex queries to the scheduler. Details of the commands to be implemented are given below in commands section. (A word about the last command, FLUSH. Suppose a job has sufficient resources but due to its low priority, other jobs with high priority keeps running ahead of it. This would lead job starvation. You give a push to such jobs to run. A way to handle this could be by artificially raising such jobs' priority.)

2. **Scheduler Interfaces**

You are required to implement the modified scheduler interface mentioned below also attached in **ProjectManagement.zip**. This is designed to be backward compatible, and an implementation of the original scheduler interface (for Assignment 4) also automatically implements the new interface. Well, almost. You will still need to include the interfaces and default implementations for JobReport_ and UserReport_.

Note that new methods in the interface have the timed prefix. These methods will be timed by the driver code. (You are welcome to modify your driver code accordingly.) Please make sure that the timed methods return as soon as possible with the correct action/answer. This means avoid all extraneous statements (like print) that are not required.

You may only import java.util.ArrayList, java.util.Stack, and java.util.Queue. (Maps are not allowed for assignment 5. Maps will be allowed for assignment 4 tests only.)

You can download **ProjectManagement.zip** from:
ProjectManagement.zip

```
package ProjectManagement;

import java.util.ArrayList;

/**
 * DO NOT MODIFY
 */
public interface SchedulerInterface {
    /**
     * @param cmd Handles Project creation. Input is the command from
INP1 file in array format (use space to split it)
     */
    void handle_project(String[] cmd);

    /**
     * @param cmd Handles Job creation. Input is the command from INP1
file in array format (use space to split it)
     */
    void handle_job(String[] cmd);

    /**
     * @param name Handles user creation
     */
    void handle_user(String name);

    /**
     * Returns status of a job
     *
     * @param key
     */
    void handle_query(String key);

    /**
     * Next cycle, is executed whenever an empty line is found.
     */
    void handle_empty_line();

    /**
     * Add budget to a project Input is the command from INP1 file in array
format (use space to split it)
     *
     * @param cmd
     */
    void handle_add(String[] cmd);


    /**
     * If there are no lines in the input commands, but there are jobs which
can be executed, let the system run till there are no jobs left (which can be
run).
     */
    void run_to_completion();

    /**
```

```
     * After execution is done, print the stats of the system
     */
    void print_stats();

    // Timed queries for the old queries. These are equivalent to their untimed
parts.
    // Only they should not print anything so the real code is timed.
    default void timed_handle_user(String name){}
    default void timed_handle_job(String[] cmd){}
    default void timed_handle_project(String[] cmd){}
    default void timed_run_to_completion(){}

    //------------ New queries---------
    /*
     * In the format below, <> enclose parameter.
     * Format: PROJECT <PROJECT> <T1> <T2> =>
     *    Return list of all Jobs for project <PROJECT> arriving at <T1> or later
and at <T2> or earlier
     *
     * Format: USER <USER> <T1> <T2>  =>
     *    Return list of all Jobs of user <USER> arriving at <T1> or later and at
<T2> or earlier
     *
     * Format: PROJECTUSER <PROJECT> <USER> <T1> <T2>  =>
     *    Return list of all Jobs of user <USER> for project <PROJECT>
arriving at <T1> or later and at <T2> or earlier.
     *    This list must be sorted in the order of job completion. Unfinished
Jobs come last, and in the order of their arrival.
     *
     * Format: PRIORITY <PRIORITY> =>
     *     Return the list of waiting (unfinished) Jobs with a priority higher than
or equal to <PRIORITY>
     *
     */
    default ArrayList<JobReport_> timed_report(String[] cmd){ return null;}

   /*
    * Return the list of top <top> budget consuming users (cumulative usage)
    *    (sorted by consumption first, and then by the user's latest job's
completion time)
    *    Note that budget is consumed when the job is finished, not when it is
waiting.
    */
    default ArrayList<UserReport_>timed_top_consumer(int top){return null;}

   /*
    * "Prioritize" long waiting jobs: Execute all jobs waiting for <waittime> or
longer,
    *    if there is sufficient budget, in the order of their relative priority
    */
    default void timed_flush(int waittime){ }
}
```

3. **Additional Interfaces**

```
interface JobReport_ {
   default String user(); { return null; }
   default String project_name()  { return null; }
   default int budget()  { return 0; }
   default int arrival_time()  { return 0; }
   default int completion_time() { return 0; }
}

interface UserReport_ {
   default String user();     { return null; }
   default int consumed(); { return 0; }
}
```

4. **Timer.**

Here is one way to time execution of a function.

```
public class Timer {
  private long startTime = System.currentTimeMillis();
  private long elapsedTime = 0;

  public void start() { startTime = System.currentTimeMillis(); }
  public long since() { elapsedTime = System.currentTimeMillis() -
startTime; return elapsedTime;}
  public long report() { return elapsedTime; }
  public void report(String s) { System.out.println(s + elapsedTime); }
}
```

4. Sample input file:

1. USER Rob
2. USER Harry
3. USER Carry
4. PROJECT IITD.CS.ML.ICML 10 15
5. PROJECT IITD.CS.OS.ASPLOS 9 100
6. PROJECT IITD.CS.TH.SODA 8 100
7. JOB DeepLearning IITD.CS.ML.ICML Rob 10
8. JOB ImageProcessing IITD.CS.ML.ICML Carry 10
9. JOB Pipeline IITD.CS.OS.ASPLOS Harry 10
10. JOB Kmeans IITD.CS.TH.SODA Carry 10
11.
12. QUERY Kmeans
13. QUERY Doesnotexists
14.
15. JOB DeepLearningNoProject IITD.CS.ML.ICM Rob 10
16. JOB DeepLearningNoUser IITD.CS.ML.ICML Rob2 10
17.
18. JOB DeepLearning1 IITD.CS.ML.ICML Rob 10
19. JOB ImageProcessing1 IITD.CS.ML.ICML Carry 10
20. JOB Pipeline1 IITD.CS.OS.ASPLOS Rob 10
21. JOB Kmeans1 IITD.CS.TH.SODA Carry 10
22.
23. JOB DeepLearning2 IITD.CS.ML.ICML Rob 10
24. JOB ImageProcessing2 IITD.CS.ML.ICML Carry 10
25. JOB Pipeline2 IITD.CS.OS.ASPLOS Harry 10
26. JOB Kmeans2 IITD.CS.TH.SODA Carry 10
27.
28. ADD IITD.CS.ML.ICML 60
29. JOB DeepLearning3 IITD.CS.ML.ICML Rob 10
30. JOB ImageProcessing3 IITD.CS.ML.ICML Carry 10
31. JOB Pipeline3 IITD.CS.OS.ASPLOS Harry 10
32. JOB Kmeans3 IITD.CS.TH.SODA Carry 10
33.
34. QUERY Kmeans
35.
36. JOB DeepLearning4 IITD.CS.ML.ICML Rob 10
37. JOB ImageProcessing4 IITD.CS.ML.ICML Carry 10
38. JOB Pipeline4 IITD.CS.OS.ASPLOS Harry 10
39. JOB Kmeans4 IITD.CS.TH.SODA Carry 10
40.
41. JOB DeepLearning5 IITD.CS.ML.ICML Rob 10
42. JOB ImageProcessing5 IITD.CS.ML.ICML Carry 10
43. JOB Pipeline5 IITD.CS.OS.ASPLOS Harry 10
44. JOB Kmeans5 IITD.CS.TH.SODA Carry 10
45.
46. QUERY Kmeans
47.
48. NEW_PROJECT IITD.CS.ML.ICML 1 100
49. NEW_USER Rob 1 100

```
50. NEW_PROJECTUSER IITD.CS.ML.ICML Rob 1 100
51. NEW_PRIORITY 5
52. NEW_TOP 4
53. NEW_FLUSH 30
```

5. Sample output file:

```
 1. Creating user
 2. Creating user
 3. Creating user
 4. Creating project
 5. Creating project
 6. Creating project
 7. Creating job
 8. Creating job
 9. Creating job
10. Creating job
11. Running code
12. Remaining jobs: 4
13. Executing: DeepLearning from: IITD.CS.ML.ICML
14. Project: IITD.CS.ML.ICML budget remaining: 5
15. Execution cycle completed
16. Querying
17. Kmeans: NOT FINISHED
18. Querying
19. Doesnotexists: NO SUCH JOB
20. Running code
21. Remaining jobs: 3
22. Executing: ImageProcessing from: IITD.CS.ML.ICML
23. Un-sufficient budget.
24. Executing: Pipeline from: IITD.CS.OS.ASPLOS
25. Project: IITD.CS.OS.ASPLOS budget remaining: 90
26. Execution cycle completed
27. Creating job
28. No such project exists. IITD.CS.ML.ICM
29. Creating job
30. No such user exists: Rob2
31. Running code
32. Remaining jobs: 1
33. Executing: Kmeans from: IITD.CS.TH.SODA
34. Project: IITD.CS.TH.SODA budget remaining: 90
35. Execution cycle completed
36. Creating job
37. Creating job
38. Creating job
39. Creating job
40. Running code
41. Remaining jobs: 4
42. Executing: DeepLearning1 from: IITD.CS.ML.ICML
43. Un-sufficient budget.
44. Executing: ImageProcessing1 from: IITD.CS.ML.ICML
45. Un-sufficient budget.
46. Executing: Pipeline1 from: IITD.CS.OS.ASPLOS
47. Project: IITD.CS.OS.ASPLOS budget remaining: 80
48. Execution cycle completed
49. Creating job
50. Creating job
51. Creating job
52. Creating job
53. Running code
54. Remaining jobs: 5
55. Executing: DeepLearning2 from: IITD.CS.ML.ICML
56. Un-sufficient budget.
57. Executing: ImageProcessing2 from: IITD.CS.ML.ICML
```

```
 58. Un-sufficient budget.
 59. Executing: Pipeline2 from: IITD.CS.OS.ASPLOS
 60. Project: IITD.CS.OS.ASPLOS budget remaining: 70
 61. Execution cycle completed
 62. ADDING Budget
 63. Creating job
 64. Creating job
 65. Creating job
 66. Creating job
 67. Running code
 68. Remaining jobs: 11
 69. Executing: ImageProcessing from: IITD.CS.ML.ICML
 70. Project: IITD.CS.ML.ICML budget remaining: 55
 71. Execution cycle completed
 72. Querying
 73. Kmeans: COMPLETED
 74. Running code
 75. Remaining jobs: 10
 76. Executing: DeepLearning1 from: IITD.CS.ML.ICML
 77. Project: IITD.CS.ML.ICML budget remaining: 45
 78. Execution cycle completed
 79. Creating job
 80. Creating job
 81. Creating job
 82. Creating job
 83. Running code
 84. Remaining jobs: 13
 85. Executing: ImageProcessing1 from: IITD.CS.ML.ICML
 86. Project: IITD.CS.ML.ICML budget remaining: 35
 87. Execution cycle completed
 88. Creating job
 89. Creating job
 90. Creating job
 91. Creating job
 92. Running code
 93. Remaining jobs: 16
 94. Executing: DeepLearning2 from: IITD.CS.ML.ICML
 95. Project: IITD.CS.ML.ICML budget remaining: 25
 96. Execution cycle completed
 97. Querying
 98. Kmeans: COMPLETED
 99. Running code
100. Remaining jobs: 15
101. Executing: ImageProcessing2 from: IITD.CS.ML.ICML
102. Project: IITD.CS.ML.ICML budget remaining: 15
103. Execution cycle completed
104. Project query
105. Time elapsed (ns): 1313755
106. User query
107. Time elapsed (ns): 457942
108. Project User query
109. Time elapsed (ns): 508123
110. Priority query
111. Time elapsed (ns): 252904
112. Top query
113. Time elapsed (ns): 979585
114. Flush query
115. Job{user='Carry', project='IITD.CS.TH.SODA', jobstatus=REQUESTED,
     execution_time=10, end_time=null, priority=2147483647, name='Kmeans1'}
116. Job{user='Carry', project='IITD.CS.TH.SODA', jobstatus=REQUESTED,
     execution_time=10, end_time=null, priority=2147483647, name='Kmeans2'}
117. Job{user='Rob', project='IITD.CS.ML.ICML', jobstatus=REQUESTED,
     execution_time=10, end_time=null, priority=2147483647,
     name='DeepLearning3'}
```

```
118. Job{user='Carry', project='IITD.CS.ML.ICML', jobstatus=REQUESTED,
     execution_time=10, end_time=null, priority=2147483647,
     name='ImageProcessing3'}
119. Job{user='Harry', project='IITD.CS.OS.ASPLOS', jobstatus=REQUESTED,
     execution_time=10, end_time=null, priority=2147483647, name='Pipeline3'}
120. Job{user='Carry', project='IITD.CS.TH.SODA', jobstatus=REQUESTED,
     execution_time=10, end_time=null, priority=2147483647, name='Kmeans3'}
121. Job{user='Rob', project='IITD.CS.ML.ICML', jobstatus=REQUESTED,
     execution_time=10, end_time=null, priority=10, name='DeepLearning4'}
122. Job{user='Carry', project='IITD.CS.ML.ICML', jobstatus=REQUESTED,
     execution_time=10, end_time=null, priority=10, name='ImageProcessing4'}
123. Job{user='Harry', project='IITD.CS.OS.ASPLOS', jobstatus=REQUESTED,
     execution_time=10, end_time=null, priority=9, name='Pipeline4'}
124. Job{user='Carry', project='IITD.CS.TH.SODA', jobstatus=REQUESTED,
     execution_time=10, end_time=null, priority=8, name='Kmeans4'}
125. Job{user='Rob', project='IITD.CS.ML.ICML', jobstatus=REQUESTED,
     execution_time=10, end_time=null, priority=10, name='DeepLearning5'}
126. Job{user='Carry', project='IITD.CS.ML.ICML', jobstatus=REQUESTED,
     execution_time=10, end_time=null, priority=10, name='ImageProcessing5'}
127. Job{user='Harry', project='IITD.CS.OS.ASPLOS', jobstatus=REQUESTED,
     execution_time=10, end_time=null, priority=9, name='Pipeline5'}
128. Job{user='Carry', project='IITD.CS.TH.SODA', jobstatus=REQUESTED,
     execution_time=10, end_time=null, priority=8, name='Kmeans5'}
129. Running code
130. Remaining jobs: 14
131. Executing: Kmeans1 from: IITD.CS.TH.SODA
132. Project: IITD.CS.TH.SODA budget remaining: 80
133. Execution cycle completed
134. Running code
135. Remaining jobs: 13
136. Executing: Kmeans5 from: IITD.CS.TH.SODA
137. Project: IITD.CS.TH.SODA budget remaining: 70
138. Execution cycle completed
139. Running code
140. Remaining jobs: 12
141. Executing: Pipeline5 from: IITD.CS.OS.ASPLOS
142. Project: IITD.CS.OS.ASPLOS budget remaining: 60
143. System execution completed
144. Running code
145. Remaining jobs: 11
146. Executing: ImageProcessing5 from: IITD.CS.ML.ICML
147. Project: IITD.CS.ML.ICML budget remaining: 5
148. System execution completed
149. Running code
150. Remaining jobs: 10
151. Executing: DeepLearning5 from: IITD.CS.ML.ICML
152. Un-sufficient budget.
153. Executing: Kmeans4 from: IITD.CS.TH.SODA
154. Project: IITD.CS.TH.SODA budget remaining: 60
155. System execution completed
156. Running code
157. Remaining jobs: 8
158. Executing: Pipeline4 from: IITD.CS.OS.ASPLOS
159. Project: IITD.CS.OS.ASPLOS budget remaining: 50
160. System execution completed
161. Running code
162. Remaining jobs: 7
163. Executing: ImageProcessing4 from: IITD.CS.ML.ICML
164. Un-sufficient budget.
165. Executing: DeepLearning4 from: IITD.CS.ML.ICML
166. Un-sufficient budget.
167. Executing: Kmeans3 from: IITD.CS.TH.SODA
168. Project: IITD.CS.TH.SODA budget remaining: 50
169. System execution completed
170. Running code
```

```
Remaining jobs: 4
Executing: Pipeline3 from: IITD.CS.OS.ASPLOS
Project: IITD.CS.OS.ASPLOS budget remaining: 40
System execution completed
Running code
Remaining jobs: 3
Executing: ImageProcessing3 from: IITD.CS.ML.ICML
Un-sufficient budget.
Executing: DeepLearning3 from: IITD.CS.ML.ICML
Un-sufficient budget.
Executing: Kmeans2 from: IITD.CS.TH.SODA
Project: IITD.CS.TH.SODA budget remaining: 40
System execution completed
--------------STATS---------------
Total jobs done: 19
Job{user='Rob', project='IITD.CS.ML.ICML', jobstatus=COMPLETED, execution_time=10, end_time=10, priority=10, name='DeepLearning'}
Job{user='Harry', project='IITD.CS.OS.ASPLOS', jobstatus=COMPLETED, execution_time=10, end_time=20, priority=9, name='Pipeline'}
Job{user='Carry', project='IITD.CS.TH.SODA', jobstatus=COMPLETED, execution_time=10, end_time=30, priority=8, name='Kmeans'}
Job{user='Rob', project='IITD.CS.OS.ASPLOS', jobstatus=COMPLETED, execution_time=10, end_time=40, priority=9, name='Pipeline1'}
Job{user='Harry', project='IITD.CS.OS.ASPLOS', jobstatus=COMPLETED, execution_time=10, end_time=50, priority=9, name='Pipeline2'}
Job{user='Carry', project='IITD.CS.ML.ICML', jobstatus=COMPLETED, execution_time=10, end_time=60, priority=10, name='ImageProcessing'}
Job{user='Rob', project='IITD.CS.ML.ICML', jobstatus=COMPLETED, execution_time=10, end_time=70, priority=10, name='DeepLearning1'}
Job{user='Carry', project='IITD.CS.ML.ICML', jobstatus=COMPLETED, execution_time=10, end_time=80, priority=10, name='ImageProcessing1'}
Job{user='Rob', project='IITD.CS.ML.ICML', jobstatus=COMPLETED, execution_time=10, end_time=90, priority=10, name='DeepLearning2'}
Job{user='Carry', project='IITD.CS.ML.ICML', jobstatus=COMPLETED, execution_time=10, end_time=100, priority=10, name='ImageProcessing2'}
Job{user='Carry', project='IITD.CS.TH.SODA', jobstatus=COMPLETED, execution_time=10, end_time=110, priority=2147483647, name='Kmeans1'}
Job{user='Carry', project='IITD.CS.TH.SODA', jobstatus=COMPLETED, execution_time=10, end_time=120, priority=8, name='Kmeans5'}
Job{user='Harry', project='IITD.CS.OS.ASPLOS', jobstatus=COMPLETED, execution_time=10, end_time=130, priority=9, name='Pipeline5'}
Job{user='Carry', project='IITD.CS.ML.ICML', jobstatus=COMPLETED, execution_time=10, end_time=140, priority=10, name='ImageProcessing5'}
Job{user='Carry', project='IITD.CS.TH.SODA', jobstatus=COMPLETED, execution_time=10, end_time=150, priority=8, name='Kmeans4'}
Job{user='Harry', project='IITD.CS.OS.ASPLOS', jobstatus=COMPLETED, execution_time=10, end_time=160, priority=9, name='Pipeline4'}
Job{user='Carry', project='IITD.CS.TH.SODA', jobstatus=COMPLETED, execution_time=10, end_time=170, priority=2147483647, name='Kmeans3'}
Job{user='Harry', project='IITD.CS.OS.ASPLOS', jobstatus=COMPLETED, execution_time=10, end_time=180, priority=2147483647, name='Pipeline3'}
Job{user='Carry', project='IITD.CS.TH.SODA', jobstatus=COMPLETED, execution_time=10, end_time=190, priority=2147483647, name='Kmeans2'}
------------------------
Unfinished jobs:
Job{user='Rob', project='IITD.CS.ML.ICML', jobstatus=REQUESTED, execution_time=10, end_time=null, priority=10, name='DeepLearning5'}
Job{user='Carry', project='IITD.CS.ML.ICML', jobstatus=REQUESTED, execution_time=10, end_time=null, priority=10, name='ImageProcessing4'}
Job{user='Rob', project='IITD.CS.ML.ICML', jobstatus=REQUESTED, execution_time=10, end_time=null, priority=10, name='DeepLearning4'}
Job{user='Carry', project='IITD.CS.ML.ICML', jobstatus=REQUESTED, execution_time=10, end_time=null, priority=2147483647, name='ImageProcessing3'}
```

```
211. Job{user='Rob', project='IITD.CS.ML.ICML', jobstatus=REQUESTED,
     execution_time=10, end_time=null, priority=2147483647,
     name='DeepLearning3'}
212. Total unfinished jobs: 5
213. --------------STATS DONE---------------
```

Schedular_Driver_Assignment5.zip

6. Submission instructions
As always compress src directory to zip format and rename the zip file in the format entry number assignment5.zip. For example, if your entry number is 2012CSZ8019, the zip file should be named 2012CSZ8019 assignment5.zip. Then you need to convert this zip file to base64 format as follows and submit the b64 file on Moodle.

```
base64 entrynumber_assignment5.zip > entrynumber_assignment5.zip.b64
```

7. Folder structure
Inside the src directory, you need to have a README.txt or README.pdf (case sensitive) and your solution (exactly following the folder structure of the code provided in assignment 4.). Please do not rename the existing directories.

8. MOSS
Please note that we will run MOSS on the submitted code. Anyone found with a copied code, either from the Internet or from another student, will be dealt as per the class policy

Note again:

- You have to build your own driver code
- Maps (e.g. HashMap) are not allowed

Updates !

- Please download the updated userReport and JobReport(12/10/2019). You have to replace the older one from your ProjectManagement.zip folder.It is usefull to make your interface backword compatible with Test4.
  UserReport_JobReport.zip
- Please download the updated input_output file here(12/10/2019)
  input_output_updated.zip
- Updated driver code and sample input/output specification is availabe. (08/10/2019)
  Schedular_Driver_Assignment5.zip
- You are allowed to import list, and linked list! (08/10/2019)
- A small change in ProjectManagement.zip! please use latest one (06/10/2019)
  ProjectManagement.zip